Université de Strasbourg

Master CSMI-2025

Study of the relationship between road surface composition and their ability to reflect light.

Author:
Alexis GONIN

Project supervisor:
Valérie MUZET

August 28, 2025

Abstract

I completed my M1 CSMI internship at CEREMA Strasbourg as part of the ENDSUM research team. To optimise road lighting, the knowledge of the photometric properties of road surface is essential. Since measurements are in practice rarely done, the development of a prediction model based on the material properties and its age would be very useful for lighting designers. To achieve this goal, a collection of data was collected internationally within the CIE TC4-50 and with the Cerema COLUROUTE measuring device. It comprises excel files with metadata on the description of the road surface (composition, localisation and age,...) and the measured photometric data.

During this internship, starting from Excel tables, a database was created to format, clean all the data and add new standard features. Data visualization methods were then implemented, enabling preliminary and graphical analyses, both to represent the database and identify the features that can be used for modelling.

Subsequently, machine learning models like random forest and boosted tree were applied to predict specularity index of road surfaces based on their composition. Model interpretation techniques were also developed, allowing the identification of factors influencing the reflective capacity of road surfaces. Finally, to move beyond complex models, a linear model based on the interpretations of these complex models was proposed. The classification performance of all these models was compared in the case of three specularity clusters. The entire process has been made available on Cerema's Gitlab and will be available for use in the context of the work of TC4-50 of the CIE.

Contents

I Introduction . . . 3

I.1 CEREMA . . . 3

I.2 Context of the internship . . . 4

I.3 Objectives . . . 5

II Road photometry . . . 5

II.1 Photometric characteristics of a pavement surface . . . 5

II.2 Photometric classification of road surfaces . . . 7

III Creation and completion of the database . . . 9

III.1 Input data presentation . . . 9

III.2 MongoDB database . . . 11

III.3 Input data formatting and cleaning . . . 11

III.4 New features . . . 13

III.5 Database description . . . 17

III.6 Example of an entry in the database . . . 18

IV Display and visual analysis of the data . . . 20

IV.1 Age effect on photometric properties . . . 20

IV.2 Study of different asphalts concretes . . . 21

IV.3 Impact of surface treatments on specularity . . . 23

IV.4 Impact of granular colour and granular level of lightness . . . 24

IV.5 Conclusion on the data visual analysis . . . 26

V Basis of machine learning . . . 27

V.1 Decision tree vs deep learning comparison . . . 27

V.2 Bagging algorithm and Random Forests . . . 27

V.2.1 Bagging . . . 27

V.2.2 Random Forests . . . 28

V.2.3 Out of bag approach 29
V.3 Histogram-based Gradient Boosting 30
VI Models application 31
VI.1 Objectives of machine learning 31
VI.2 one-hot encoding 31
VI.3 Python Implementation 32
VI.4 Results 32
VII Interpreting Model Predictions 35
VII.1 Permutation Feature Importance 36
VII.2 Shap Values 38
VII.2.1 Theory 38
VII.2.2 Applying SHAP values to our regression analysis 39
VII.3 Construction of a model using mean SHAP values with induced impact 42
VIII Work environment and code transmission 45
IX Conclusion and perspectives 45
Bibliography 48
Appendix 49
I New features graphic 49
II Dataset example 50
III Extract of keys, values and repartition 51
IV Visual analysis 57
V SHAP Values 59
VI Code user guide 60

I Introduction

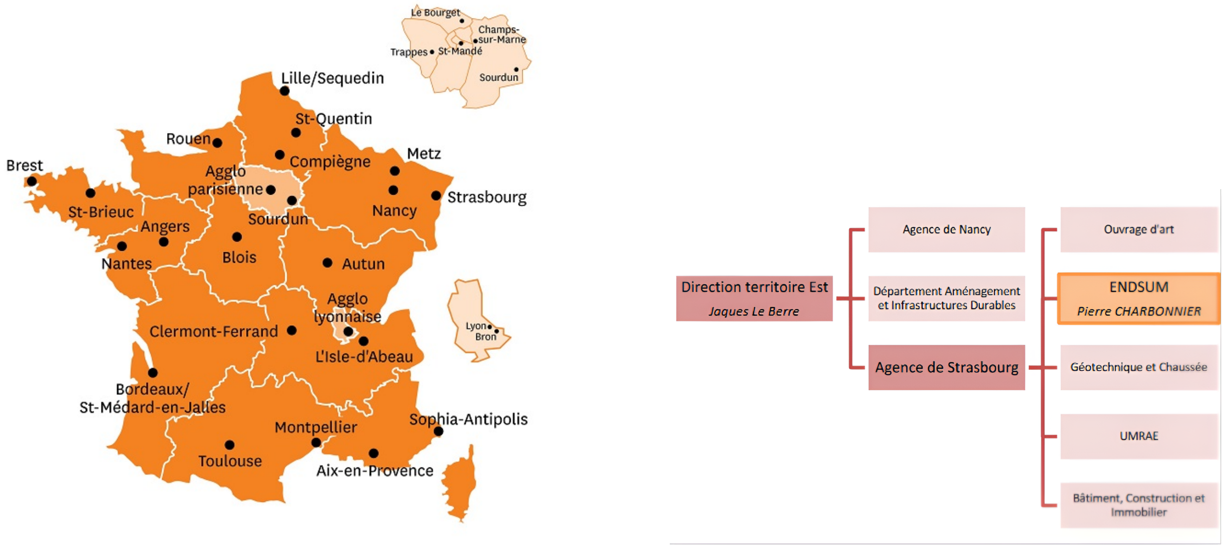
In our ever-evolving society, the importance of mobility cannot be overstated. As our lives become increasingly interconnected and fast-paced, safe and efficient road infrastructures have become essential. One key aspect of road safety is road visibility which ensures that drivers can navigate their way safely, particularly during challenging conditions or by night. During nighttime, visibility of obstacles is influenced by two essential factors: the illumination from luminaires and the reflection of light from the road surfaces. While luminaires are responsible for providing light on the road, the reflective properties of the road surfaces themselves significantly impact visibility. The interaction between the amount of light emitted by the luminaires and the light reflected by the road surfaces directly affects how clearly drivers can see obstacles at night. Consequently, comprehending and optimizing the reflective properties of road surfaces become integral aspects of road infrastructure design. The issue is complex because these properties depend on both internal factors [1] linked to the composition (type of road surface, used aggregates and binders,...) and application of road surfaces and also external factors [2] such as the effect of the age of the surface, traffic or the impact of the environmental conditions to which it is exposed. Moreover, measurements are seldom done. It is within this context and in cooperation with foreign partners that Cerema has accumulated a large number of photometric measurements on a wide range of road surfaces. Once this data has been gathered, the challenge is to be able to make use of it.

I.1 CEREMA

This internship will be carried out at CEREMA. Cerema (Center for Studies and Expertise on Risks, Environment, Mobility, and Urban Planning) was founded on January 1, 2014. It is a public institution under the supervision of the Ministry for the Ecological and Inclusive Transition and the Ministry for Territorial Cohesion, with the objective of promoting a transition towards a resource-efficient, decarbonized, environmentally friendly, and equitable economy. Cerema is present throughout mainland France and the overseas territories, with 26 locations and 2,400 staff members.

The internship was completed at the Strasbourg agency of the Eastern Territorial Directorate, more specifically within the ENDSUM research team (Non-Destructive Evaluation of Structures and Materials). The department consists of 7 permanent employees and hosts 3 PhD students.

This team specializes in various fields: image processing, photometry, artificial intelligence, and 3D. These methods are applied in several areas: civil engineering structures, autonomous vehicles, roads and intelligent transportation systems, the environment (cliffs), and public lighting. The work focused on the characterization of the light reflection capabilities of roads.



(a) CEREMA's implantation

(b) Organization chart

Figure 1: CEREMA's implantation and organization chart

I.2 Context of the internship

This work is part of the ANR REFLECTIVITY research project, which aims to provide industry stakeholders and local authorities with a representative database of the optical properties of road surfaces, along with new tools to measure these properties and predict their evolution. The ultimate goal is to support the optimization of public lighting systems, the reduction of the effects of urban heat islands, and the adaptation of safety strategies in the context of the emergence of automated vehicles. [3]

To optimize a lighting installation, it is essential to understand the photometry of road surfaces. However, taking into account the optical properties of road surfaces in development or maintenance projects remains challenging for local authorities and road managers.

It is possible to characterize surface photometry either in the laboratory or on site, using equipment designed to measure the amount of light reflected under specific lighting angles defined by the CIE (International Commission on Illumination). These measurements generate what is known as an "r- table," which can be used to calculate global descriptors representing surface brightness and the overall amount of reflected light.

A significant data set that comprises measurements and formulations of various road surfaces from around the world has already been collected with laboratory gonioreflectometers by the CIE Technical Committee 4-50 "Road Surface Characterization for Lighting Applications" [4]. More data were also collected on field with the Cerema COLUROUTE measuring device [5] These data will serve as the basis for the project.

I.3 Objectives

The aim of this internship is to build a database and then implement data analysis methods in order to create a classification of road surface compositions.

The internship will contain:

- Literature review and familiarisation with the dataset and topic [6] [3] [7]
- Development of a structured database to facilitate data handling, using Python-based scripts
- Exploratory data analysis to identify links between surface composition/formulation and photo-metric properties
- Implementation of a predictive method for photometric data in connection with several clustering possibilities.

II Road photometry

II.1 Photometric characteristics of a pavement surface

The photometric characteristics of a pavement surface are determined by measuring the luminance coefficient q under various lighting and observation directions [8] [7]. This luminance coefficient q (in $cd.m^{-2}.lx^{-1}$) is defined as the ratio of luminance L (in $cd.m^{-2}$) to illuminance E (in lx) as follows (Equation 1):

$$q(\alpha, \beta, \epsilon) = \frac{L(\alpha, \beta, \epsilon)}{E(\beta, \epsilon)} \tag{1}$$

A set of q values constitutes a partial BRDF (Bidirectional Reflectance Distribution Function). In road application, this coefficient is measured as a function of three angles (as illustrated in Figure 2): angle if incidence ϵ, azimuth angle β and observation angle α. By convention, $\alpha = 1°$ according to the CIE. This angle represents the view of a driver whose eyes are positioned 1.50m above the road surface looking 86m ahead, corresponding to typical conditons for driving outside urban areas.

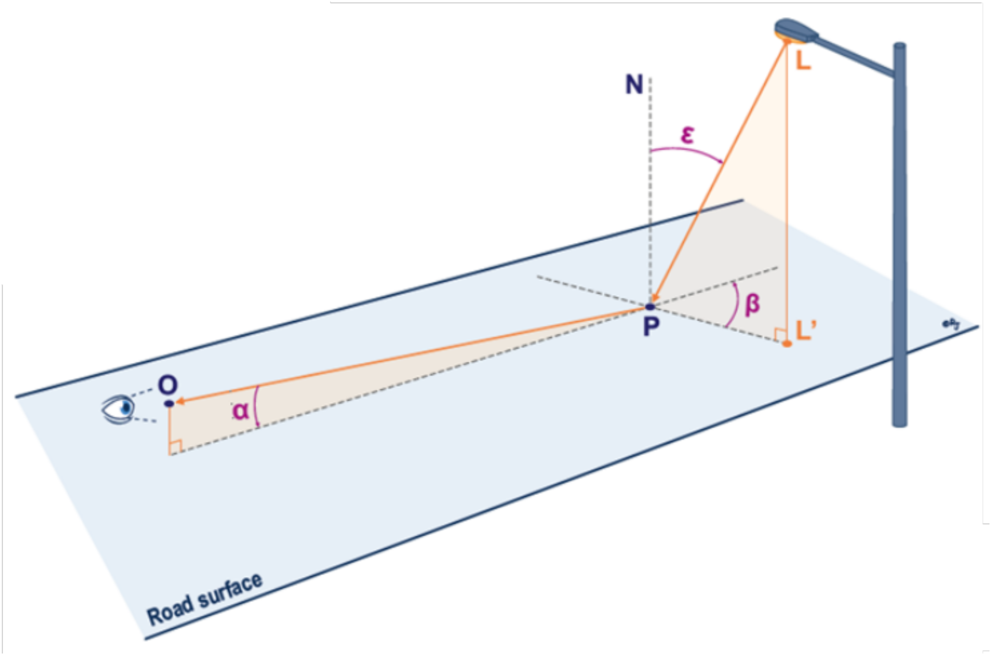


Figure 2: The luminance coefficient q depends on the lighting and observation directions, defined by the angles $\alpha = 1°$, β and ϵ

Instead of the luminance coefficient q, lighting engineers use the reduced luminance coefficient, denoted r, which is defined as a function of the luminance coefficient q and the lighting angles β and ϵ (Equation 2):

$$r(\beta, \epsilon) = q(\beta, \epsilon) \cdot \cos(\beta)^3 \tag{2}$$

The CIE has defined a reflection table [8], known as r-table, which corresponds to 580 lighting directions resulting from various combinations of the angles β and ϵ, where β ranges from 0° to 180°, and $tan(\epsilon)$ ranges from 0 to 12. In practice, 392 angle combinations are measured.

$\tan y \backslash \beta$	0	2	5	10	15	20	25	30	35	40	45	60	75	90	105	120	135	150	165	180
0.00	140	142	140	142	140	140	140	142	140	140	140	140	138	140	140	140	140	140	140	140
0.25	234	232	232	231	231	224	210	204	195	186	180	152	133	118	106	99.2	94.0	90.5	85.5	87.2
0.50	362	362	356	342	323	298	272	247	227	198	182	138	106	88.7	77.0	65.4	61.0	58.1	55.2	55.2
0.75	536	532	514	477	416	364	312	262	219	181	155	101	73.2	59.3	51.2	45.3	41.9	39.5	38.4	37.2
1.00	708	716	704	615	503	384	295	233	181	147	126	69.0	50.9	41.8	37.2	31.8	30.0	28.2	26.3	26.3
1.25	838	858	812	659	508	359	236	174	135	101	75.9	46.8	34.7	29.8	26.2	22.7	21.3	20.6	19.8	19.8
1.50	994	920	883	680	431	261	174	124	91.1	64.8	52.0	31.9	24.6	21.2	19.6	17.3	16.2	15.7	15.1	14.5
1.75	986	999	858	586	389	199	134	89.9	65.3	44.7	37.6	22.8	17.9	15.7	14.3	13.0	12.1	12.1	11.6	11.2
2.00	980	910	810	495	267	148	86.8	62.9	42.5	31.2	25.1	16.7	13.4	12.0	11.3	10.2	9.4	9.4	9.4	9.1
2.50	756	736	626	337	145	72.2	46.4	30.8	23.3	18.3	15.3	10.3	8.5	7.8	7.3	6.8	6.3	6.3	6.3	6.3
3.00	574	516	457	186	76.7	42.5	27.2	17.4	14.0	11.3	9.4	6.9	5.8	5.4	4.9	4.7	4.5	4.5	4.5	4.5
3.50	439	396	321	114	45.2	26.2	15.6	11.7	9.0	7.5	6.6	4.9	4.3	3.8	3.6	3.4	3.3	3.3	3.4	3.4
4.00	335	307	229	90.3	29.8	18.5	11.0	8.2	6.6	5.4	4.8	3.7	3.2	2.9	2.8	2.7	2.6	2.6	2.6	2.6
4.50	260	248	205	56.8	20.9	13.1	7.8	5.9	5.0	4.3	3.7	2.8	2.5	2.3	2.2	2.1	2.1	2.0	2.1	2.1
5.00	204	186	143	35.0	14.2	8.2	5.7	4.4	3.7	3.2	2.9	2.2	2.0	1.8	1.7	1.7	1.6	1.6	1.7	1.7

$\tan y \backslash \beta$	0	2	5	10	15	20	25	30	35	40	45	60	75	90	105	120	135	150	165	180
5.50	161	147	106	27.6	10.2	6.2	4.4	3.4	2.9	2.6										
6.00	132	120	102	28.1	8.4	5.1	3.5	2.9	2.5											
6.50	109	104	75.5	16.2	6.6	3.7	2.7	2.2												
7.00	93.9	89.4	47.0	12.2	5.6	3.0	2.3	1.9												
7.50	79.7	77.2	44.4	10.8	4.6	2.6	1.9													
8.00	69.6	68.3	42.3	8.6	3.9	2.1	1.6													
8.50	61.1	60.2	32.0	8.7	3.3	2.2	1.5													
9.00	55.3	55.0	27.5	6.1	2.9	1.8														
9.50	49.3	49.9	28.8	5.2	2.5	1.3														
10.00	44.9	45.7	27.4	4.4	2.2	1.2														
12.00	32.4	34.2	17	4.5	1.4															

Table 1: R table example

Based on this table, three descriptors of the pavement's optical properties can be computed.

The specularity factor S1, which reflects the glossiness of the material being studied, is defined as in Equation 3:

$$S_1 = \frac{r(\beta = 0, tan(\epsilon) = 2)}{r(\beta = 0, tan(\epsilon) = 0)} \tag{3}$$

The average luminance coefficient Q_0 (in sr^{-1} or cd.m^{-2}. lx^{-1}) is defined as the mean of the luminance coefficients over the solid angle Ω_0 which only depends on the specific values of β and $tan(\epsilon)$ defined by the r-table format (Equation 4).

$$Q_0 = \frac{1}{\Omega_0} \int q(\beta, tan(\epsilon)) \cdot sin(\epsilon) d\epsilon d\beta \tag{4}$$

The diffuse luminance coefficient Q_d (also in sr^{-1} or cd.m^{-2}. lx^{-1}) is used for daylight illumination (Equation 5):

$$Q_d = \frac{1}{\pi} \int q(\beta, tan(\epsilon)) \cdot cos(\epsilon) \cdot sin(\epsilon) d\epsilon d\beta \tag{5}$$

Note that Q_0 and Q_d are computed using the trapezoidal integration methodology [9]

II.2 Photometric classification of road surfaces

Few r-tables are measured, both because of managers' reluctance to core existing pavements and because commercial measuring equipment are not available on site, particularly for the grazing angle of 1° [10].

Since S$_1$, Q$_0$ and r-table are difficult to determine without apropriate tools, so standard tables have been introduced by the CIE [8] to facilitate lighting calculations, based on a classification of road surfaces according to the S_1 value (Table 2).

Class		**Standard r-table**			
Name	**S1 range**	**Name**	**Qd**	**Q0**	**S1**
RI	$S_1 < 0.42$	R1	0.087	0.100	0.25
RII	$0.42 \leq S_1 < 0.85$	R2	0.057	0.070	0.58
RIII	$0.85 \leq S_1 < 1.35$	R3	0.050	0.070	1.11
RIV	$1.35 \leq S_1$	R4	0.052	0.080	1.55

Table 2: Current classification and standard r-table

However, this classification is now being questioned, and several studies suggest that the standard r-tables are no longer suitable for road surfaces [11] [12]. This is why a clustering analysis is conducted in the CIE 4-50 technical committee to establish updated specularity thresholds and standard r-tables more representative of nowadays pavements. Since this analysis is still in process and the thresholds not yet validated, S1 prediction models will be developed during my internship, which will enable to adapt the implemented method to the final thresholds. However, in the graphical representations, it is possible to display S1 thresholds on demand.

The thresholds used in the graphical representations (in section IV) are 0.47 for the New RI class, 1.1 for New RII and 3.41 for New RIII (see table). They were established on the realistic data of the TC4-50 database (629 r-tables), using a clustering method based on deviation between two r-tables calculated with normalized root mean square deviation ([13]). The input data and new threshold are represented on Figure 3 and correspond to Table 4.

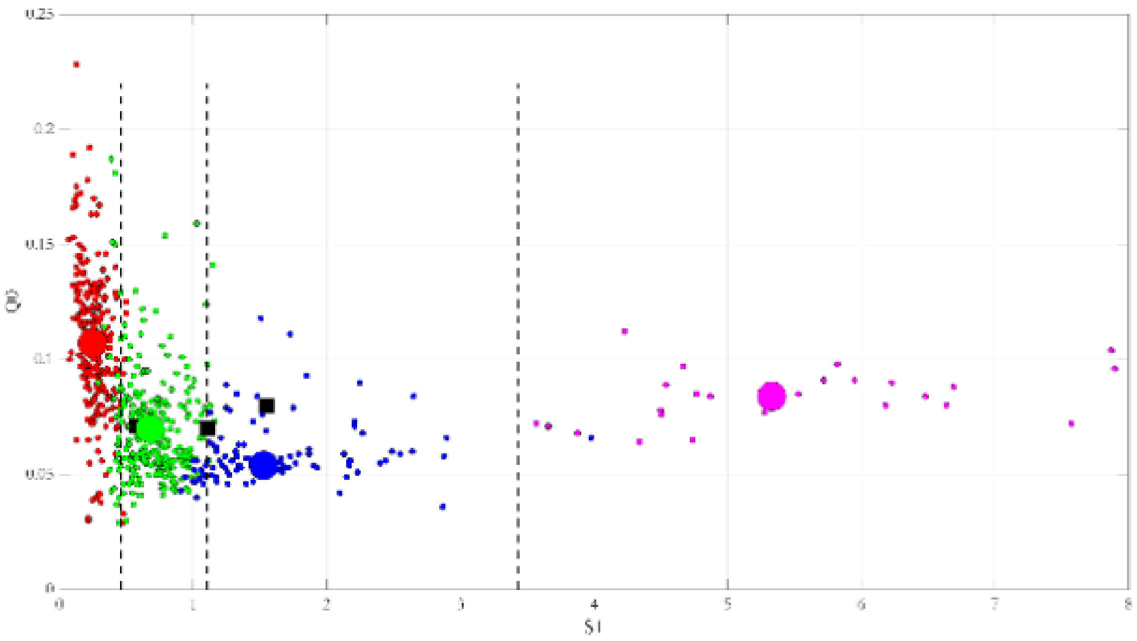


Figure 3: Representation of all Q0 in function of S1 for all TC4-50 realistic data. The ≪ New R ≫ classification is represented in color (red for class I, green for class II, blue for class III and pink for class IV) while the corresponding typical new tables are in circles. The previous typical r-tables are represented in black squares.

Table 3: Classification et valeurs standards pour les r-tables

Class	Range	Standard r-table Name	Q0	S1
New RI	S1 $<$ 0,47	New R1	0,107	0,25
New RII	0,47 $\leq$ S1 $<$ 1,10	New R2	0,070	0,68
New RIII	1,10 $\leq$ S1 $<$ 3,43	New R3	0,054	1,53
New RIV	3,43 $\leq$ S1	New R4	0,084	5,33

Table 4: New classification and standard r-tables

III Creation and completion of the database

III.1 Input data presentation

The used data originates from Excel sheets, which contain a variety of metadata and photometric data to characterize road surface samples. An example of a data sheet is provided in Annexes (Figure 24).

In these sheets, there are photometric data, which are Q_0 (4), Q_d (5) and S_1 (3) values, as well as r_tables.

To go along with these data, there are also metadata that describe the composition of the road surface, the context of the measurements and the localisation of the road (Table 5).

Type of information	**Feature**
	Mesure Name
General Information	Country of measurement Type of measure: laboratory / on site Apparatus used Date of measurement Contact person Name Contact person Email
Description of the road surface	Type of road surface Type of sample: manufactured / extracted / on site Granular: type Granular: size Granular: color Granular Other Type of binder Surface treatment Surface state: dry / moist / wet Age of road surface when measured Type of road Type of road: trafic Type of road: according to CIE 115 Transversal Localisation Localisation (City) Name of the road Is it a largelly used road surface in his type of road today? Is it innovative, experimental yet? Picture (if available) Comments

Table 5: Metadata fields

All these metadata were provided by the various people who performed the measurements around the world, with sometimes different nomenclatures depending on the country. For the sake of uniformity, standardized metadata have been created (Table 6).

Type of information	**Feature**
Updated nomenclature	General family Standard Type of road surface Granular gradation Granular continuity Binder Type of surface treatment Age of road surface when measured in months Reflection indicatrix realistic?

Table 6: Updated nomenclature

Therefore, the objective of this project is, first, to consolidate the metadata, and then to analyze the database in order to study how the different metadata parameters influence the photometric data.

You can find a table of useful keys and values associated to these key, along side their number and repartition in annex III

III.2 MongoDB database

When dealing with datasets where entries do not have relationships with each other, using MongoDB [14] [15] is particularly advantageous. As a document-oriented NoSQL database, MongoDB provides a flexible schema that allows each entry to be stored as an independent document, without the need for predefined tables or strict data models. This approach simplifies data storage and retrieval, as there is no need to manage foreign keys or complex joins. Additionally, MongoDB's architecture is optimized for scalability and high performance, making it well-suited for handling large volumes of unstructured or semi-structured data where relational integrity is not required. This allows for faster development cycles and easier adaptation to changing data requirements.

III.3 Input data formatting and cleaning

Formatting

In order to read the data correctly, some modifications need to be made to the input data. In my case, it was necessary to slightly adjust the Excel tables to properly align the keys for the future database, and above all, to remove all Excel formulas such as "=2*J4", for example.

Once this step is completed, we can build a Read function that reads all the Excel files in a directory.

Cleaning

Once the database has been created, the data needs to be cleaned. To do this, I exported to a CSV file all the unique keys existing in the database, as well as all the unique values associated with these keys. Next, we can analyze this file to look for duplicates, typo mistakes or reading errors.

To get rid of most duplicates, I decided to convert all the strings to lowercase and remove any spaces at the beginning and end. Then, the remaining issues could be resolved individually.

For example the database had different entry for:

- sand blasting
- sand blasted
- sandblasting
- Sand Blasting

To resolve these issues, I created a table (table 7) of systematic replacements and harmonizations that I applied to the database.

systematic changes in all the database	
in	**out**
gray	grey
discontinous	discontinuous
sand blasting	sand blasted
sandblasting	sand blasted
sanded	sand blasted
shot blasting	shot blasted
waterjet	water jet
continous	continuous
poured cement	poured cement concrete
poured concrete	
aggregates alluvial	alluvial
not visible	na

Table 7: Systematic replacements applied to the database

Then I created a table (table 8) of strict string replacements.

strict string verification	
in	**out**
ac	AC
acbe	ACBE
actl	ACTL
acvtl	ACVTL
sma	SMA
sd	SD
qd	QD
>24	99

Table 8: Strict string replacements applied to the database

These strict replacement ensure that ”ac” is not replaced by ”AC” for example inside of a word like ”acoustic”.

III.4 New features

The input features present several limitations. First, some are too specific and have values associated to too few entries, which can be problematic for machine learning approaches that rely on one-hot encoding (subsection VI.2) of categorical variables. Additionally, some features are missing for a large number of entries, making their use impossible. For these reasons, we decided to create new features based on the existing ones.

The first new feature I implemented is the ”database” feature. This feature just copies the name of the folder where excel files are originally stored. This allows us to know and keep track of the origin of the data, which is useful for future analysis.

By performing keyword searches in the ”type of road” feature, we create 2 new features, environment (table 10) and usage (table 9).

usage	
in (type of road)	**out (usage)**
if [count(”,”) > 1]	na
na	na
main	main road
trunk	main road
highway	main road
lane	main road
national	main road
motorway	main road
avenue	main road
expressway	main road
federal	main road
industrial	main road
secondary	secondary roads
country	secondary roads
regional	secondary roads
local	secondary roads
rural	secondary roads
county	secondary roads
district	secondary roads
street	secondary roads
moderate	secondary roads
urban road	na
sans detection	others

Table 9: Mapping from `type of road` to `usage`

environment	
in (type of road)	**out (environment)**
plus de 2 virgules	na
na	na
tunnel	tunnel
city	urban
urban	urban
street	urban
industrial	urban
parking	urban
sidewalk	urban
cycle	urban
playground	urban
sans detection	other/inter-urban

Table 10: Mapping from `type of road` to `environment`

Note that the keyword search is performed in the order given by the tables, so if a keyword matches multiple categories, it will be assigned to the first matching category. For example, if the "type of road" is "highway country road", "usage" will be assigned to "main road" and not "secondary roads" because "main road" is listed before "secondary roads" (a graphical illustration of this processus is to be found in annex section I).

I also created "granular max(D)" and "granular min(d)" by reading the "granular gradation" feature, which describes the size of aggregates in the road, and associating "granular max(D)" with the maximum value and "granular min(d)" with the minimum value.

Completion features

Many entries do not have a value for "granular : color", so I created a new feature called "granular color completed", which is equal to "granular : color" if it is not missing. Otherwise, we look at the "granular : type" field and, using a literature search, assign the color associated with that type of aggregate (table 11)[16].

color completed	
in (granular : type)	**out (color completed)**
luxovite	white
labradorite	white
synopal	white
quartzite	white
arclyte	white
dolomite	white
dark granite	dark
steel slag	dark
granite	grey

Table 11: Mapping from `granular : type` to `color completed` [16]

With this completion, we go from 383 (28%) missing values in granular: color to 118 (8%) missing values in color completed.

Categorisation features

Some features are too specific and do not allow for generalization and cause problems due to one-hot encoded variables. For these reasons, we decided to create new features that are categorizations of these features.

For example, we use keyword searches in the "granular color completed" feature to create a new feature called "granular level of brightness" using the conversion table Table 12 (a graphical illustration of this processus is to be found in section I).

Another example is the creation of an "age categorization" feature based on the reading of "age of road surface when measured in months" and using the table of classification: Table 13.

granular level of brightness	
in	**out**
na	na
and	medium
mixture	medium
,	medium
clear	clear
light	clear
light grey	clear
white	clear
black	dark
dark	dark
le reste	medium

Table 12: Mapping from granular color completed to granular level of brightness

age (int)	**age categorisation (str)**
-1	na
0	c1 0
$0 < x \leq 5$	c2 0.1-5 months
$6 < x \leq 11$	c3 6-11 months
$12 < x \leq 17$	c4 12-17 months
$18 < x \leq 23$	c5 18-23 months
$24 < x \leq 47$	c6 24-47 months
$48 < x$	c7 >48 months

Table 13: Mapping from age (in months) to age categorisation

III.5 Database description

MongoDB stores data as documents (usually in BSON format, a binary version of JSON). Each document is a key-value data structure.

A few figures to describe the database:

- 1397 entries, consisting of 855 TC4-50 entries and 542 fields measures conducted with COLUROUTE device.

- Espace mémoire occupé: 11.76 Mo
- 2 collections: one for the photometric data and metadata, and one for the associated r_tables
- First collection: photometric data
  - 3.01 Mo in JSON formating
  - 50 unique features
  - The overall proportion of data types is: str 84.10%, int 6.85%, datetime 0.21%, float 8.85%.
- Second collection: r_tables
  - 4.41 Mo in JSON formating
  - 3 unique features
  - The overall proportion of data types is: str 50.00%, list 50.00%.

The difference between the size of the JSON files and the actual space used on disk is due to indexes, internal metadata, pre-allocated space by MongoDB to optimize insertions, as well as possible padding and fragmentation. A JSON export contains only the raw data, without indexes or internal structure, and is often compressed (with fewer spaces and no padding).

III.6 Example of an entry in the database

```
{

  "_id": {
      "$oid": "68833fc6aad6f1a3ca6195ba"
  },
  "mesure name": "rt 2005 1",
  "country of measurement": "argentina",
  "type of measure : laboratory / on site": "on site",
  "apparatus used": "lal cic reflectometer",
  "date of measurement": 2005,
  "contact person name": "p. ixtaina",
  "contact person email": "na",
  "type of road surface": "asphalt microconcrete",
  "type of sample: manufactured / extracted / on site": "on site",
  "granular : type": "na",
  "granular : size": "medium size stone",
  "granular : color": "dark",
  "granular other": "hot discontinuous application",
  "type of binder": "bituminous",
  "surface treatment": "no",
  "surface state: dry / moist / wet": "dry",
  "age of road surface when measured": "2.5 years",
  "type of road": "motorway",
  "type of road : trafic": "heavy traffic (slow lane)",
  "type of road: according to cie 115": "na",
  "transversal localisation": "wheel",
  "localisation (city)": "na",
  "name of the road": "na",
  "is it a largelly used road surface in his type of road today?": "yes",
  "is it innovative, experimental yet?": "no",
  "picture (if available)": "na",
  "comments": "slow lane, test area on the right side of the lane",
  "a (observation angle) in degree": 1,
  "qd": 0.054,
  "q0": 0.072,
  "s1": 1.14,
  "r-table (sheet)": "rt 2005 1",
  "general family": "bituminous mixture",
  "standard type of road surface": "ACVTL",
  "granular gradation": "medium",
  "granular continuity": "discontinuous",
  "binder": "bituminous",
  "type of surface treatment": "raw",
  "age of road surface when measured in months": 30,
  "reflection indicatrix realistic?": "yes",
  "granular max(D)": 10.0,
  "granular min(d)": 0.0,
  "environment": "other/inter-urban",
  "usage": "main road",
  "age categorisation": "c6 24-47 months",
  "granular color completed": "dark",
  "granular level of brightness": "3-dark",
  "granular gradation categorisation": "na",
  "database": "TC4-50"

}
```

IV Display and visual analysis of the data

I developed simple methods to access and display data. In the study of road surface reflectance, the average luminance factor Q_0 is represented in function of the specularity S_1, with my methods, it is possible to represent S_1, Q_0, and Q_d on a 2D graph, each as a function of the others. Box plot can also be used to represente the effect of various features on the photometric data. The data are represented with S1 thresholds (red dashed lines) using Greffier analysis with 4 clusters (values on table Table 4).

A complete description of the display and data access methods is available in appendix section VI and with a jupyter notebook notice in the gitlab repository.

Several representation and analysis are done in this section. They are conducted on all the database (TC4-50 and in site Coluroute data) considering only the **validated*** data.

***A survey was conducted within TC4-50 taskgroup and a validation of the r-tables was done by experts based on a graphical representation called the reflection indicatrix. The results are compiled in the feature “Is the reflection indicatrix realistic ?” with a simple yes/no possibility.**

IV.1 Age effect on photometric properties

Specularity at different ages is a very classical example. According to the literature, specularity is expected to decrease with age and to stabilize after a few years. [7] Indeed, newly constructed roads are more shiny, but over time, due to sunlight, weather conditions, and vehicle traffic, they become more diffuse and the specularity decreases.

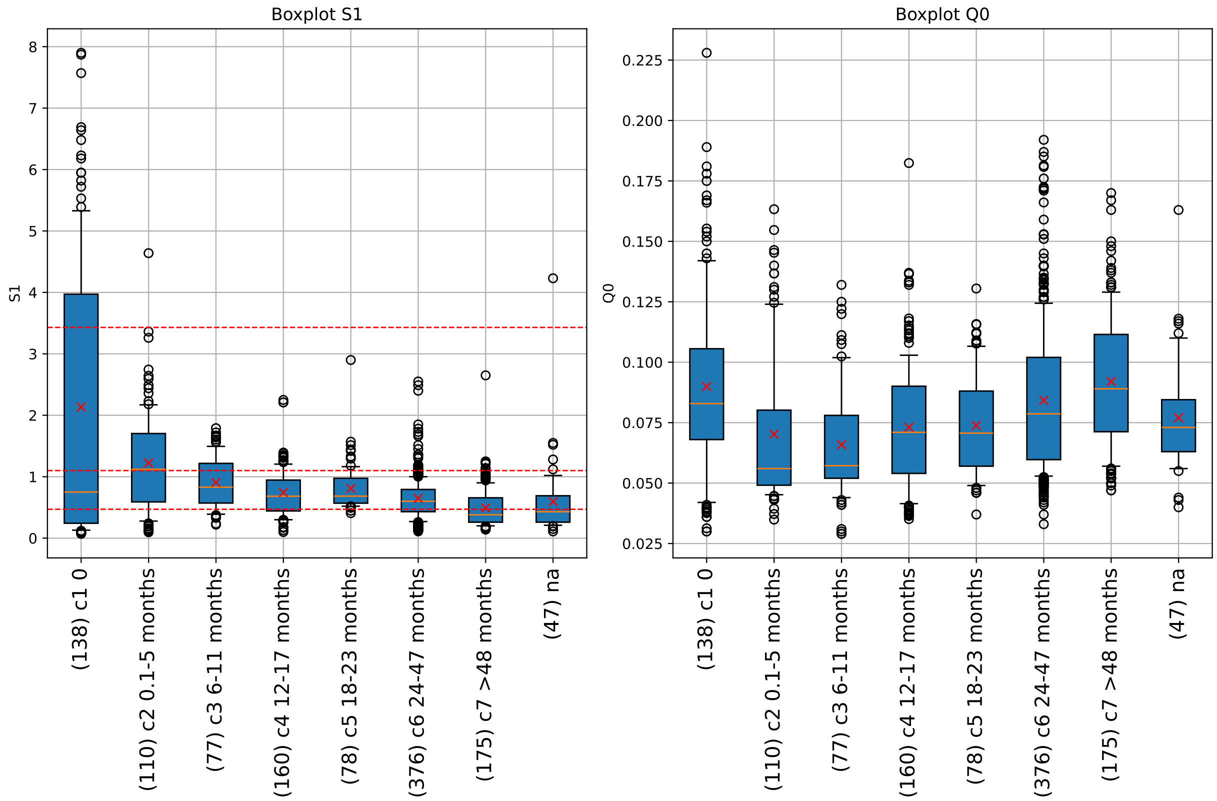


Figure 4: Specularity represented by age in months, with an example of classification threshold in red

We do indeed observe on Figure 4 the expected result: specularity decreases with age and is particularly high when the road is very young (less than one year old).

IV.2 Study of different asphalts concretes

One of the interests of this work, as mentioned earlier, is to be able to create groupings of road types by observing that their Q_0 and S_1 values are similar.

One of the questions we asked ourselves was, for example: can the different types of asphalt concrete be combined into a single category? There are 4 types of asphalt concrete (called AC for asphalt concrete, ACBE, ACTL, and ACVTL) depending of their thickness and we made the assumption that the thickness has no impact on the photometric caracteristics. [1]

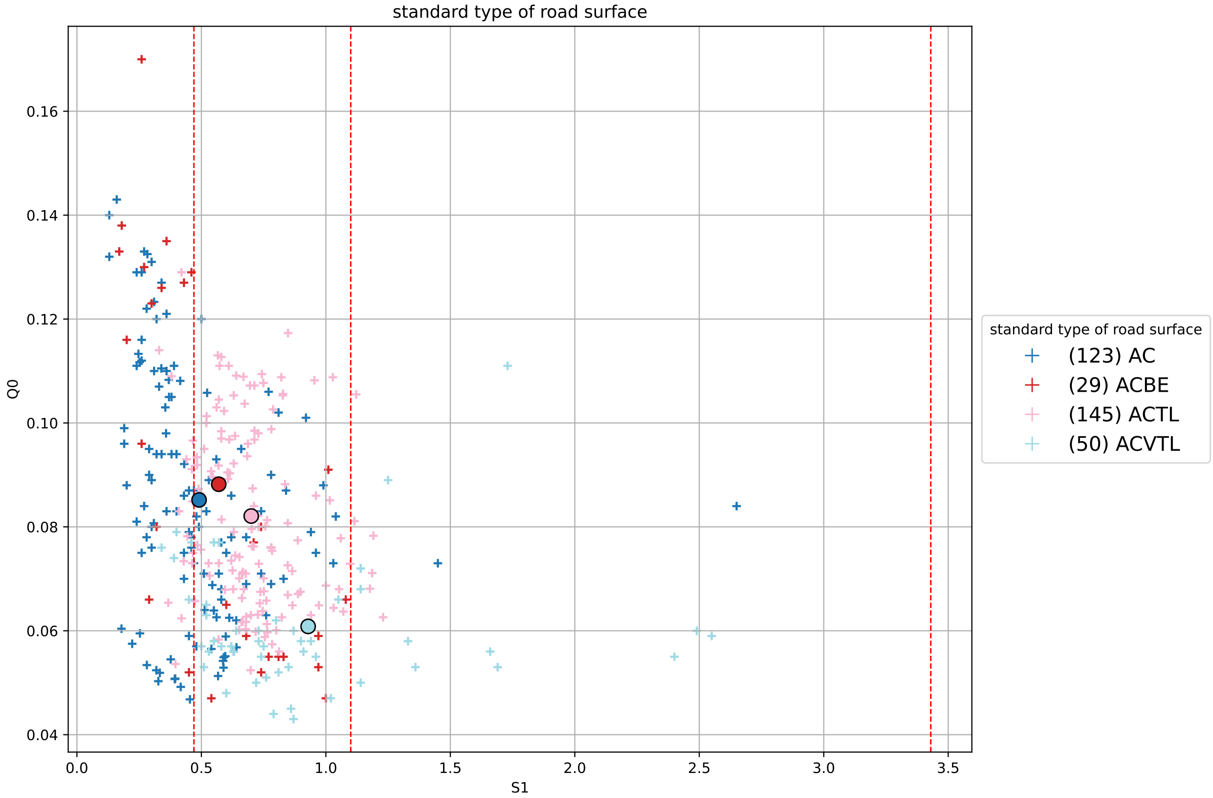


Figure 5: Asphalt concretes $Q_0 = f(S1)$, with an example of classification threshold in red

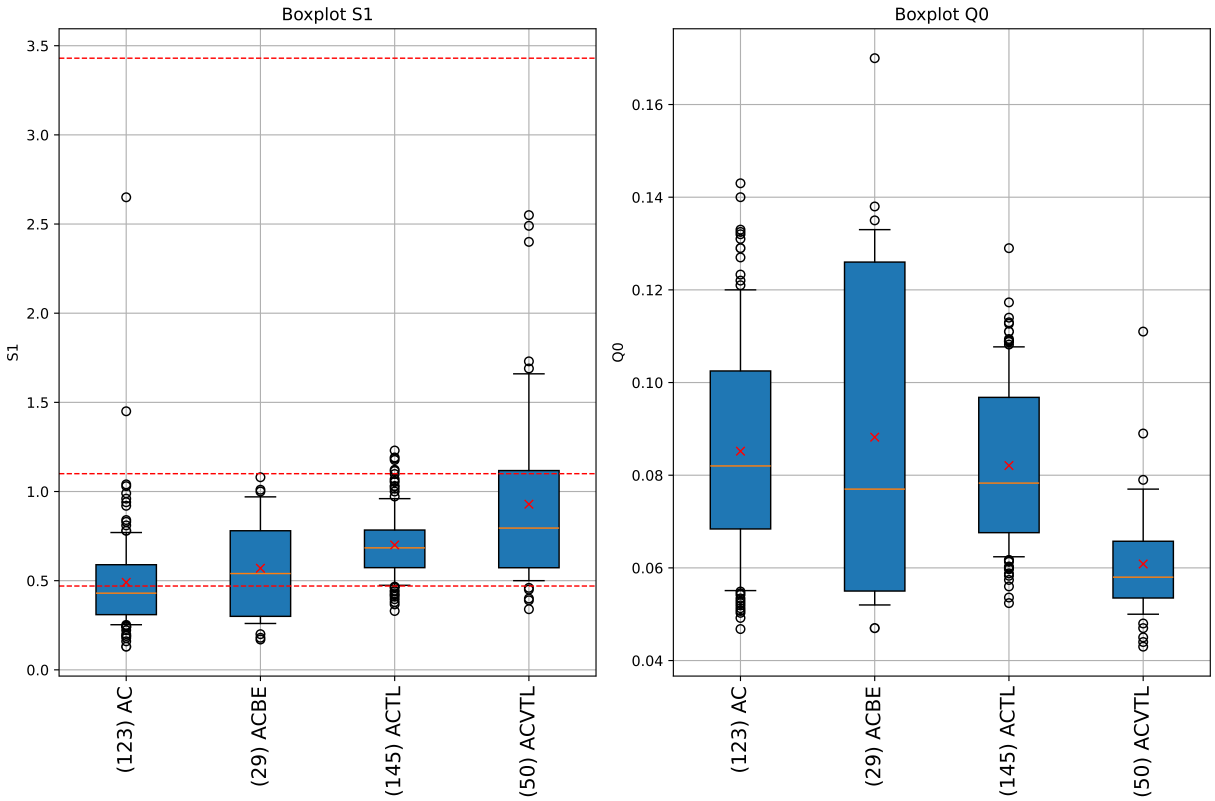


Figure 6: Asphalt concretes boxplot S1 et Q0

In Figure 5 graphs, the different kinds of asphalt concrete are represented and the large dots are the barycenters of each point cloud.

With these two representations (Figure 6, Figure 5), it seams that we can't group all the asphalt concretes together as initially hoped because ACVTL has higher specularity and lower Q0 than the others.

IV.3 Impact of surface treatments on specularity

Another expected result is the diminishing effect of surface treatments on specularity, and particularly for newly constructed roads, that is, when they are at their most specular (Figure 4). The objective of an initial road surface treatment is generally to remove the dark and specular bituminous binder and thus to decrease its initial specular effect. This is indeed observed on Figure 7, all the treated surfaces have a lower specularity than the not treated ones (called "raw" in the database).

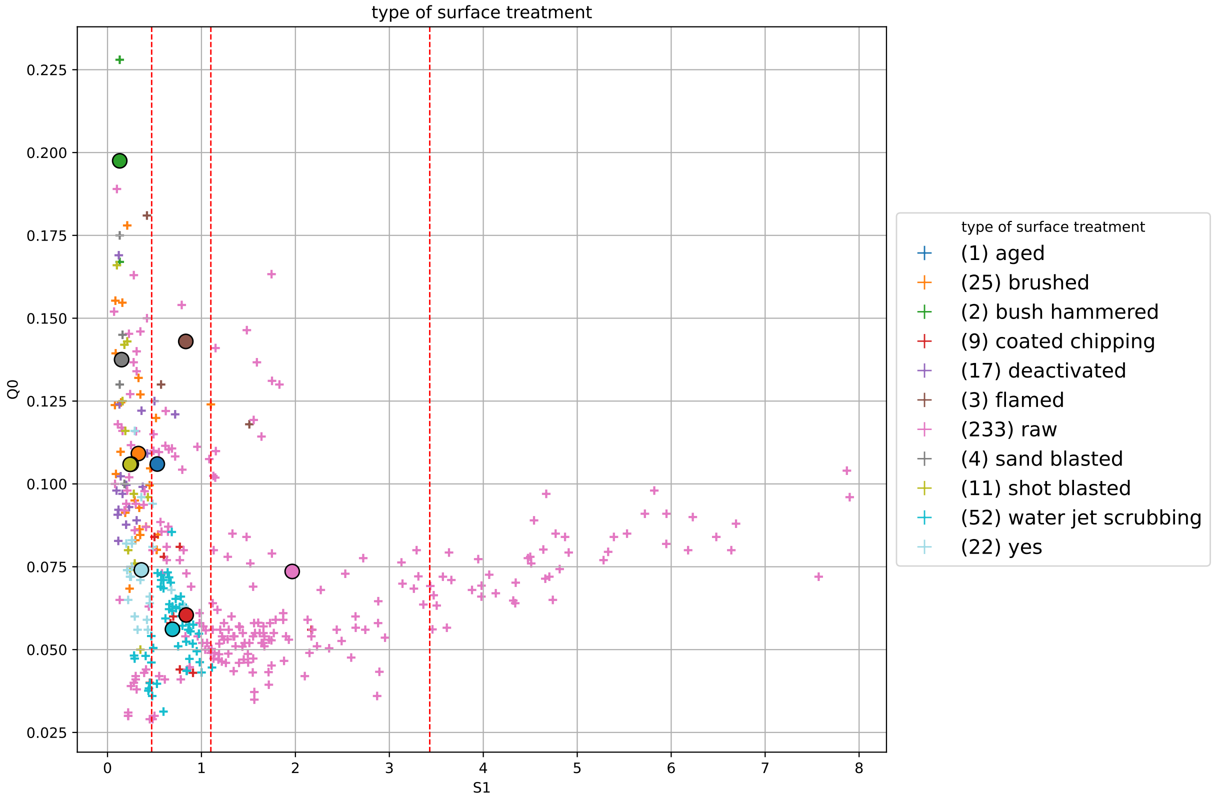


Figure 7: Impact of surface treatments on specularity for newly constructed roads (<12 months)

IV.4 Impact of granular colour and granular level of lightness

Lighter stones are generally used in the road surfaces to lighten the roads. It is known that for stabilised road surfaces, the inclusion of white or light stone result in an increase of Q0 and a decrease of S1 [12]. The effect of the granular colour and granular level of lightness of our database is studied only for stabilized pavements (aged of more than 23 months) because the binder could have an important effect when the pavements are young.

When we used the initial field "granular colours", it was quite difficult to conclude as you can see on Figure 8. Some class have few data and others have slightly different denominations.

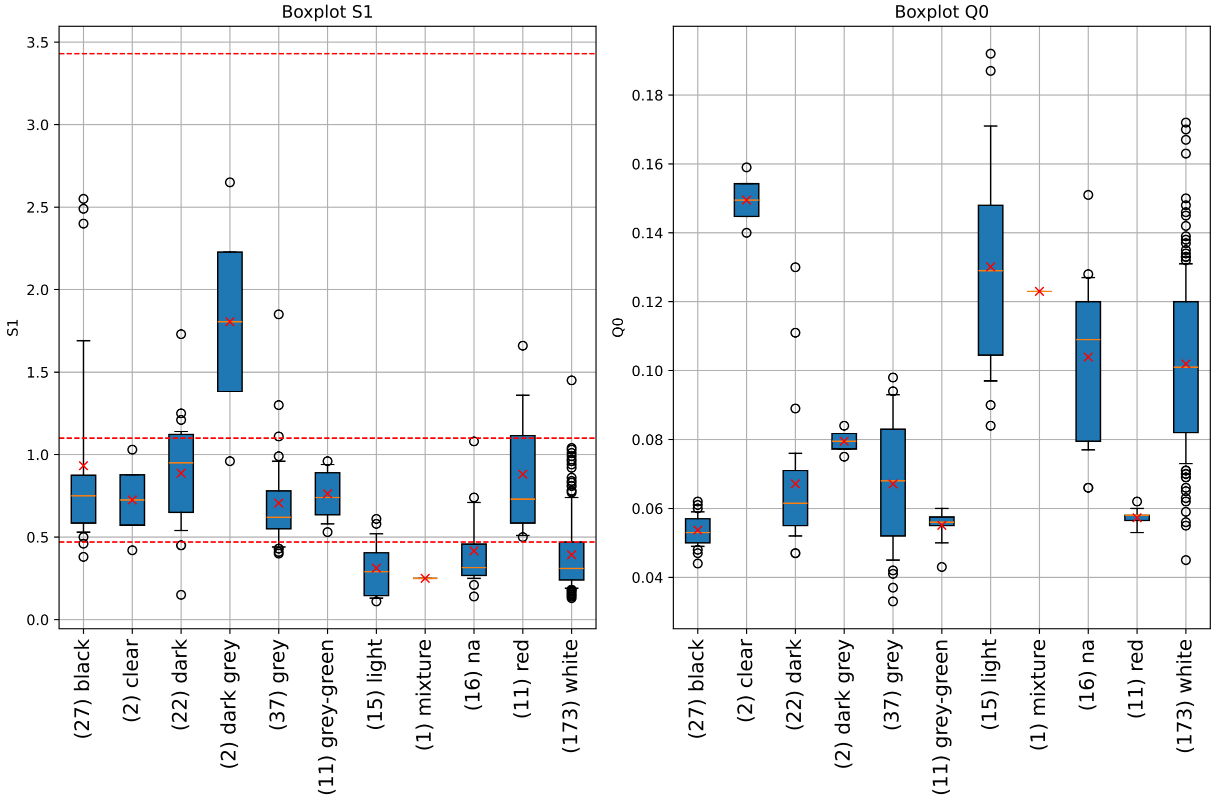


Figure 8: Impact of granular colours on S1 and Q0

The figure Figure 9 represents the barplot for S1 on the left and Q0 on the right for the "granular levels of brightness". As expected, the use of light stone in the composition of a pavements increases its Q0 and decrease the specularity S1. On the contrary, the use of dark stones generates pavements with higher specularity and the lowest Q0.

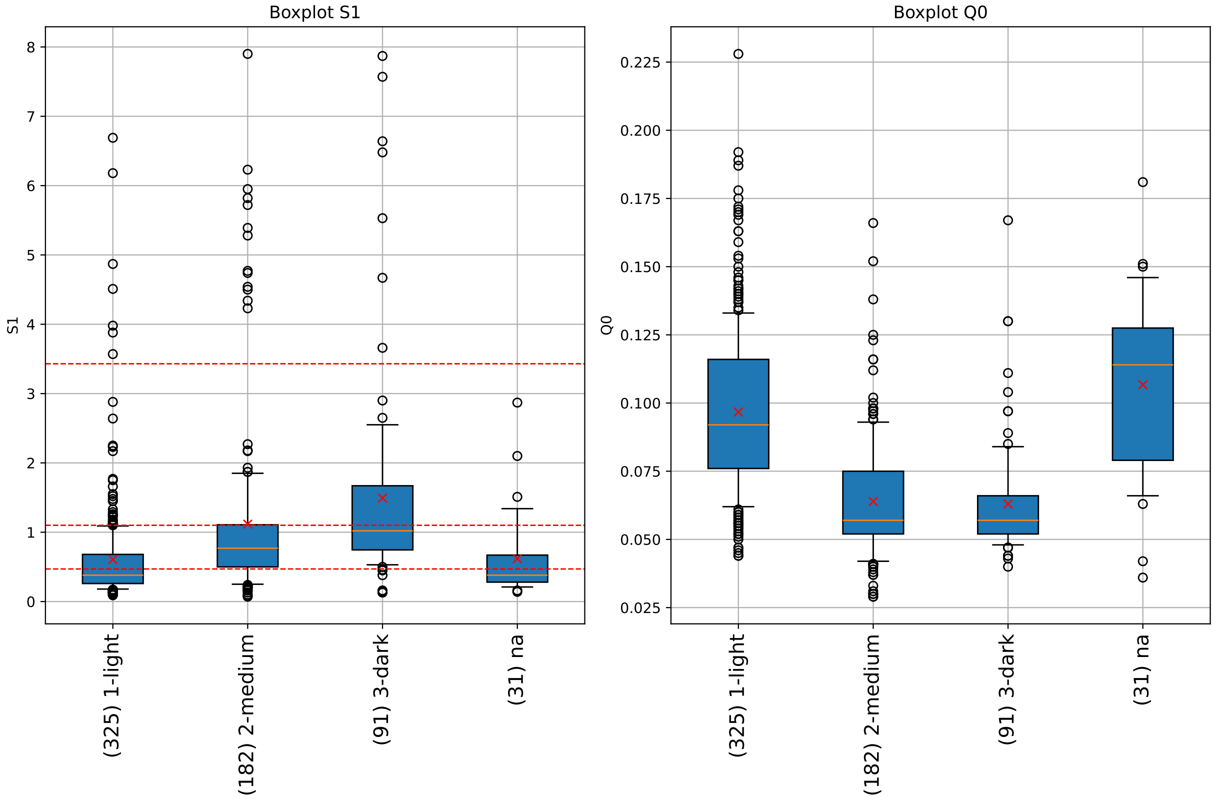


Figure 9: Impact of granular levels of brightness on S1 and Q0

IV.5 Conclusion on the data visual analysis

On this chapter some simple feature effect were represented, using the developed functions. More reprepresentations are in annex. Queries and representations of the database is really easy to do using the jupiter notice. Some more results can be found in the annex section IV

The results for the effects of age, binder, granular level of brightness, initial surface treatment are in accordance with the litterature [12] [17]. It shows that our database is consistent and our new features relevant. However, several metadata are related and the simple representation that were conducted here cannot the different consider multifactorial effect. These effects will be studied later on because they are considered in the machine learning models.

V Basis of machine learning

V.1 Decision tree vs deep learning comparison

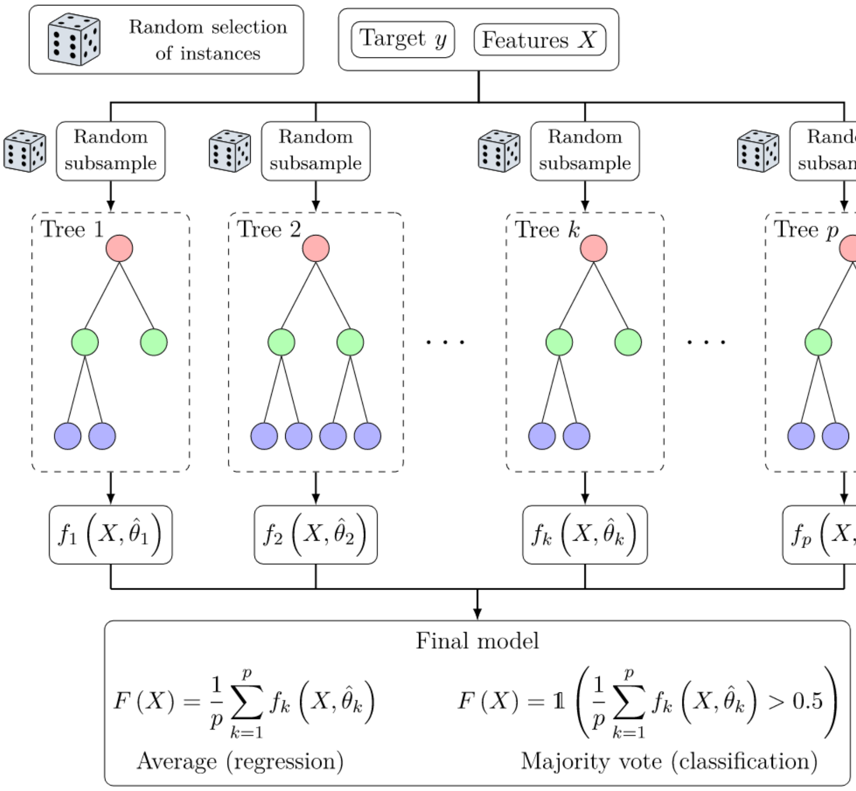
In this internship, models based on ensemble methods and decision trees were chosen. According to a literature review [18], deep learning approaches are highly effective for natural language processing, images, and audio, but their superiority is less established for applications relying on tabular data. Available comparisons conclude that tree-based ensemble methods are either more effective than deep learning approaches or perform equally well. Three advantages of ensemble methods can be identified: they are not very sensitive to irrelevant explanatory variables, robust to outliers in explanatory variables, and capable of approximating highly irregular functions. Moreover, in practice, ensemble methods are often faster to train and less resource-intensive, and hyperparameter optimization is often less complex.

Ensemble methods based on decision trees are divided into two main families, which differ in how the base models are constructed. When the base models are trained in parallel and independently, the method is called bagging (Bootstrap Aggregating). Random forest is a particularly effective variant of bagging. When the base models are trained sequentially, with each model aiming to correct the errors of the previous models, the method is called boosting.

V.2 Bagging algorithm and Random Forests

V.2.1 Bagging

Bagging (Bootstrap Aggregating) is an ensemble method that relies on aggregating the predictions of several individual models, trained independently, to build a more effective global model.



Random selection of instances
Target y
Features X
Random subsample
Random subsample
Random subsample
Random subsample
Tree 1
Tree 2
Tree k
Tree p
. . .
. . .
$f_1\left(X, \hat{\theta}_1\right)$
$f_2\left(X, \hat{\theta}_2\right)$
$f_k\left(X, \hat{\theta}_k\right)$
$f_p\left(X, \hat{\theta}_p\right)$
Final model
$F(X) = \frac{1}{p}\sum_{k=1}^{p} f_k\left(X, \hat{\theta}_k\right)$
Average (regression)
$F(X) = \mathbb{1}\left(\frac{1}{p}\sum_{k=1}^{p} f_k\left(X, \hat{\theta}_k\right) > 0.5\right)$
Majority vote (classification)

Figure 10: Schematic representation of a bagging algorithm from [18]

Figure 10 provides a schematic representation of bagging: first, random subsamples with replacement are generated from the training dataset. Then, decision trees are trained independently on these subsamples. Finally, their predictions are aggregated to obtain the final predictions. Majority voting is generally used (the class most frequently predicted by the trees) in a classification problem, and averaging in a regression problem.

V.2.2 Random Forests

Random Forests are a variant of bagging that aim to produce highly effective models by balancing two objectives: maximizing the predictive power of individual trees and minimizing the correlation between these trees (the inherent problem with bagging). To achieve this second objective, random forests introduce a new source of randomization: the random selection of variables. During the construction of each tree, instead of using all available variables to determine the best split at each node, a random subset of variables is selected. By limiting the amount of information each tree has access to at each new split, this additional step mechanically forces the trees to be more diverse (since two trees will not necessarily choose the same variables for the same splits). This significantly reduces the correlation between trees, thus improving the effectiveness of aggregation. The set of predictions thus becomes more accurate and less subject to random fluctuations.

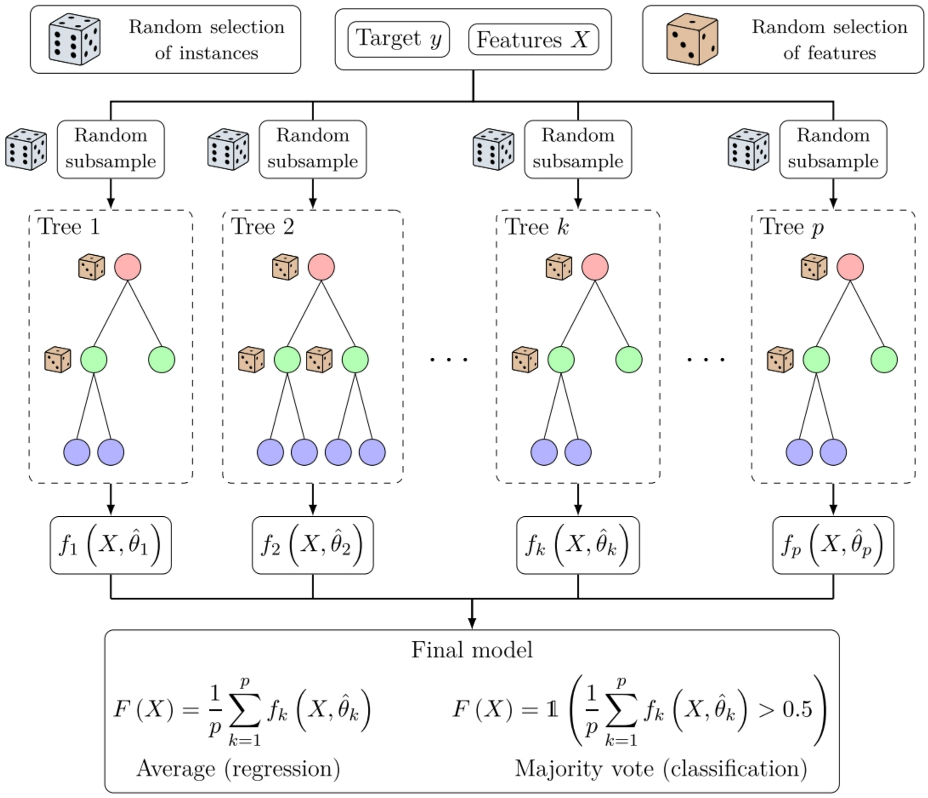

Random selection of instances
Target y
Features X
Random selection of features
Random subsample
Random subsample
Random subsample
Random subsample
Tree 1
Tree 2
Tree k
Tree p
$f_1\left(X, \hat{\theta}_1\right)$
$f_2\left(X, \hat{\theta}_2\right)$
$f_k\left(X, \hat{\theta}_k\right)$
$f_p\left(X, \hat{\theta}_p\right)$
Final model
$F(X) = \frac{1}{p}\sum_{k=1}^{p} f_k\left(X, \hat{\theta}_k\right)$
Average (regression)
$F(X) = \mathbb{1}\left(\frac{1}{p}\sum_{k=1}^{p} f_k\left(X, \hat{\theta}_k\right) > 0.5\right)$
Majority vote (classification)

Figure 11: Schematic representation of a random forest algorithm from [18]

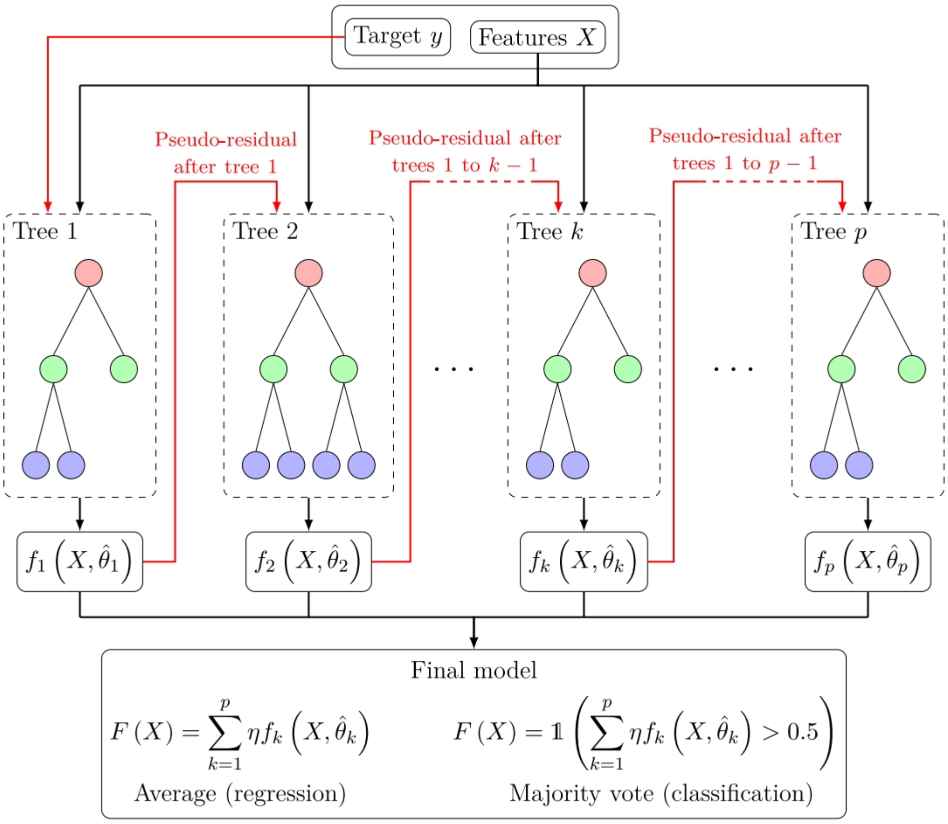
Figure 11 provides a schematic representation of a random forest. The overall logic remains the same as for bagging. Bootstrap sampling is unchanged, but the construction step for each tree is modified to use, at each new split, only a random subset of variables. The aggregation of predictions is then performed in the same way as for bagging.

V.2.3 Out of bag approach

Random forest algorithms have an interesting and very useful feature in practice: it is possible to evaluate the performance of a random forest directly from the training data, thanks to the estimation of the Out-of-Bag (OOB) error. This feature is based on the fact that each tree is built from a bootstrap sample, i.e., a sample drawn with replacement. This means that some observations are not used to train a given tree. These left-out observations form a so-called out-of-bag sample, which can be used to evaluate the performance of each tree. For each observation in the training set, a prediction can be constructed that aggregates only the predictions of the trees for which this observation is out-of-bag; this prediction is not affected by overfitting (since this observation was never used to train these trees). In this way, it is possible to properly evaluate the performance of the random forest by comparing these predictions with the target variable using a suitable metric. [18]

V.3 Histogram-based Gradient Boosting

Unlike random forests, which combine complex and independent decision trees, gradient boosting builds an ensemble of simpler trees trained sequentially. Each tree aims to correct the errors made by the previous trees, gradually improving the accuracy of the overall model.



Target y
Features X
Pseudo-residual after tree 1
Pseudo-residual after trees 1 to $k-1$
Pseudo-residual after trees 1 to $p-1$
Tree 1
Tree 2
Tree k
Tree p
. . .
. . .
$f_1\left(X,\hat{\theta}_1\right)$
$f_2\left(X,\hat{\theta}_2\right)$
$f_k\left(X,\hat{\theta}_k\right)$
$f_p\left(X,\hat{\theta}_p\right)$
Final model
$F(X)=\sum_{k=1}^{p}\eta f_k\left(X,\hat{\theta}_k\right)$
Average (regression)
$F(X)=\mathbb{1}\left(\sum_{k=1}^{p}\eta f_k\left(X,\hat{\theta}_k\right)>0.5\right)$
Majority vote (classification)

Figure 12: Schematic representation of a boosting algorithm from [18]

The logic of gradient boosting is illustrated in Figure 12:

- A first simple and not very effective model is trained on the data.
- A second model is trained to correct the errors of the first model (for example, by giving more weight to poorly predicted observations);
- This process is repeated by adding simple models, each model correcting the errors made by all previous models;

- All these models are finally combined (often by a weighted sum) to obtain a complex and effective model.

VI Models application

VI.1 Objectives of machine learning

The objective of machine learning in this internship is to create a model capable of predicting S_1 and/or the class to which a road surface belongs (subsection II.2) using the available metadata.

Another objective is to make the model understandable and interpretable, both to verify the consistency of the predictions and to deepen knowledge about the photometry of road surfaces by studying the impact attributed by the model to the different metadata.

VI.2 one-hot encoding

One Hot Encoding is a method for converting categorical variables into a binary format. It creates new columns for each category where 1 means the category is present and 0 means it is not. The primary purpose of One Hot Encoding is to ensure that categorical data can be effectively used in machine learning models.

We use one hot Encoding because:

- Eliminating Ordinality: Many categorical variables have no inherent order (e.g., "Male" and "Female"). If we were to assign numerical values (e.g., Male = 0, Female = 1) the model might mistakenly interpret this as a ranking and lead to biased predictions. One Hot Encoding eliminates this risk by treating each category independently.

- Improving Model Performance: By providing a more detailed representation of categorical variables. One Hot Encoding can help to improve the performance of machine learning models. It allows models to capture complex relationships within the data that might be missed if categorical variables were treated as single entities.

- Compatibility with Algorithms: Many machine learning algorithms, particularly those based on linear regression and gradient descent, require numerical input. One Hot Encoding ensures that categorical variables are converted into a suitable format.

In our case, we have a lot of categorical variables, so we will use One Hot Encoding to convert them into a format that can be used by machine learning algorithms. (See example in table 14).

general family	**general family_cement concrete**	**general family_surface coating**	**general family_natural material**	**general family_bituminous mixture**
cement concrete	1	0	0	0
surface coating	0	1	0	0
natural material	0	0	1	0
bituminous mixture	0	0	0	1

Table 14: Example of one-hot encoding for the feature "general family". Each category is represented by a binary vector.

VI.3 Python Implementation

In accordance with the choice made in subsection V.1, the models used are a random forest algorithm and a boosting tree algorithm. To implement these learning methods in Python, the **Scikit-learn** library [19] is used, which provides well-established implementations for Random Forest and Gradient Boosting algorithms. This library also offers tools for one-hot encoding of categorical variables, functions for cross-validation, hyperparameter optimization methods via grid search, and other utilities relevant to this work.

To manage the models, their optimization, use, evaluation, and the display of results, a specifically developed Python class described in section VI is used.

VI.4 Results

To compare the models, their respective performances are examined using the coefficient of determination R^2. All evaluations are performed on 5 folds and then averaged, except for the out-of-bag evaluation.

The models are evaluated in four situations:

- On the entire dataset. (1171 samples)
- On data from roads older than 24 months. (551 samples)
- On all data from the tc4-50 database. (629 samples)
- On data from roads older than 24 months in the tc4-50 database. (317 samples)

Both models are evaluated in two configurations for the boosting tree model and three configurations for the random forest:

- The base model, without hyperparameter optimization.
- The model after hyperparameter optimization by grid search.

- The optimized model with out-of-bag evaluation (only for random forest).

Data used for training	**Model**	**Default parameters**	**Optimised by gridsearch**	**Out-of-bag evaluation**
TC4-50	RandomForestRegressor	0,731	0,765	0,809
	HistGradientBoostingRegressor	0,716	0,716	–
All database	RandomForestRegressor	0,817	0,819	0,825
	HistGradientBoostingRegressor	0,803	0,807	–
TC4-50 $\geq$24 months	RandomForestRegressor	0,429	0,461	0,445
	HistGradientBoostingRegressor	0,443	0,458	–
All database $\geq$24 months	RandomForestRegressor	0,377	0,401	0,454
	HistGradientBoostingRegressor	0,422	0,444	–

Table 15: Comparison of R^2 performance across different datasets and model configurations

It can be observed that in almost all scenarios (except for the case of data from roads older than 24 months in the tc4-50 database), the Random Forest model outperforms the Gradient Boosting model. This result is likely explained by the use of the out-of-bag method, which allows the model to be trained on a larger amount of data.

After obtaining the regression results, classification can be performed. Table 16 provides the classification results for 3 classes with the optimized random forest model evaluated using the out-of-bag method in the best and worst scenarios.

Métrique	**All datas**	**TC4-50 + âge $\geq$ 24 mois**
R^2	0.825	0.445
Accuracy (3 classes)	0.856	0.801

Table 16: Classification results for Random Forest model with 3 classes, trained on all data and on TC4-50 dataset with age $\geq$ 24 months.

It can be seen in Table 16 that despite a low coefficient of determination, an interesting accuracy score can be achieved if the classes are sufficiently broad. To illustrate the results, Figure 13 and Figure 14 show the accuracy of the random forest in the best and worst scenarios, respectively.

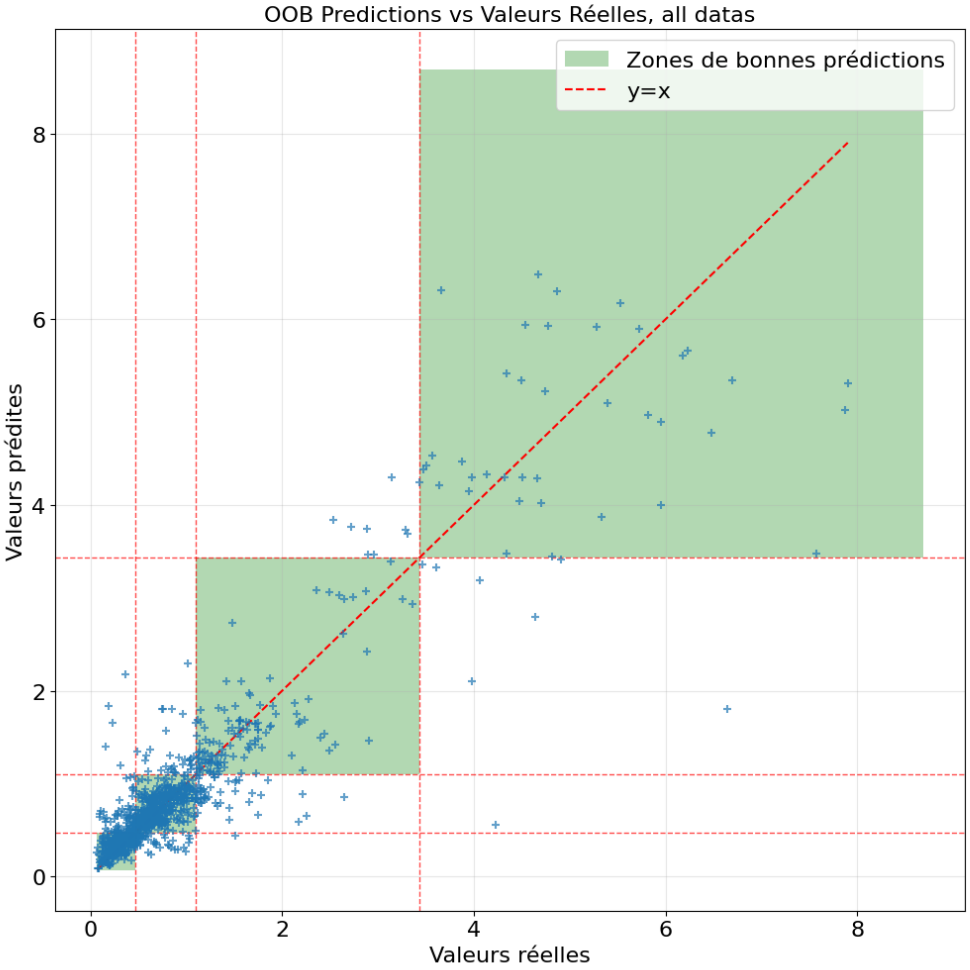


Figure 13: Visualisation of the model predictions, trained on all data. $R^2 = 0.825$, Accuracy $= 0.856$

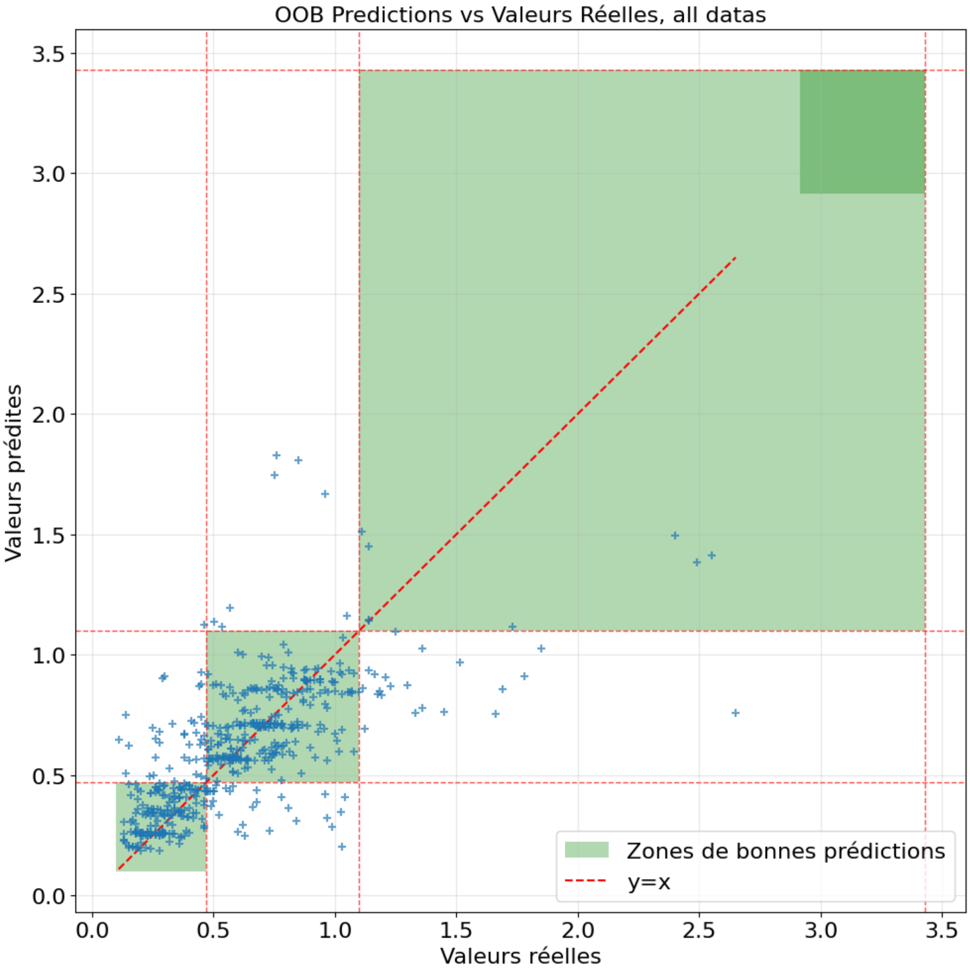


Figure 14: Visualisation of the model predictions, trained on roads older than 24 months. $R^2 = 0.445$, Accuracy $= 0.801$

VII Interpreting Model Predictions

One of the objectives of TC 4-50 is to predict the class of a pavement based on its metadata. Since, in practice, not all metadata are always available, it is important to identify which metadata have the greatest impact on the predictions of the optimal model. To study the impact of features on the predictions of the best model (to be specified), several methods have been implemented:

- A feature permutation method based on [19]
- SHAP value analysis.

The following results correspond to the realistic data from the TC4-50+ Coluroute database.

VII.1 Permutation Feature Importance

Permutation feature importance is a model inspection technique that measures the contribution of each feature to a trained model's statistical performance on a given tabular dataset. This technique is particularly useful for non-linear or opaque estimators, and involves randomly shuffling the values of a single feature and observing the resulting degradation of the model's score. By breaking the relationship between the feature and the target, we determine how much the model relies on such particular feature. [19]

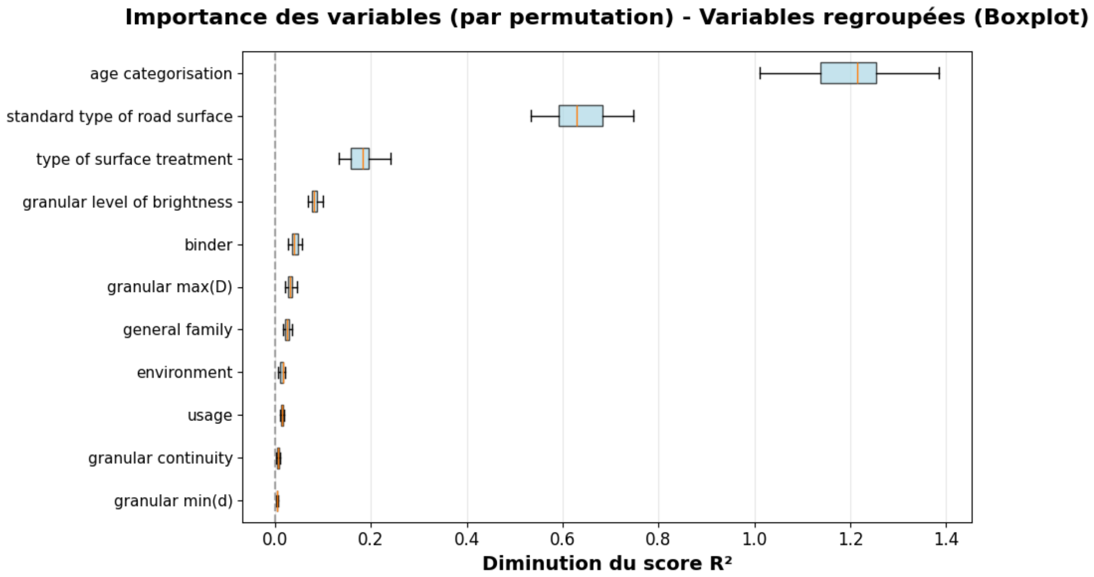


Figure 15: Permutation feature importance sorted by decreasing regression score (30 evaluations), Random Forest trained on all datas

Figure 15 illustrates the importance of each feature for the regression score of the Random Forest model trained on all ages of roads. It shows that the most important variable is the age of the road.

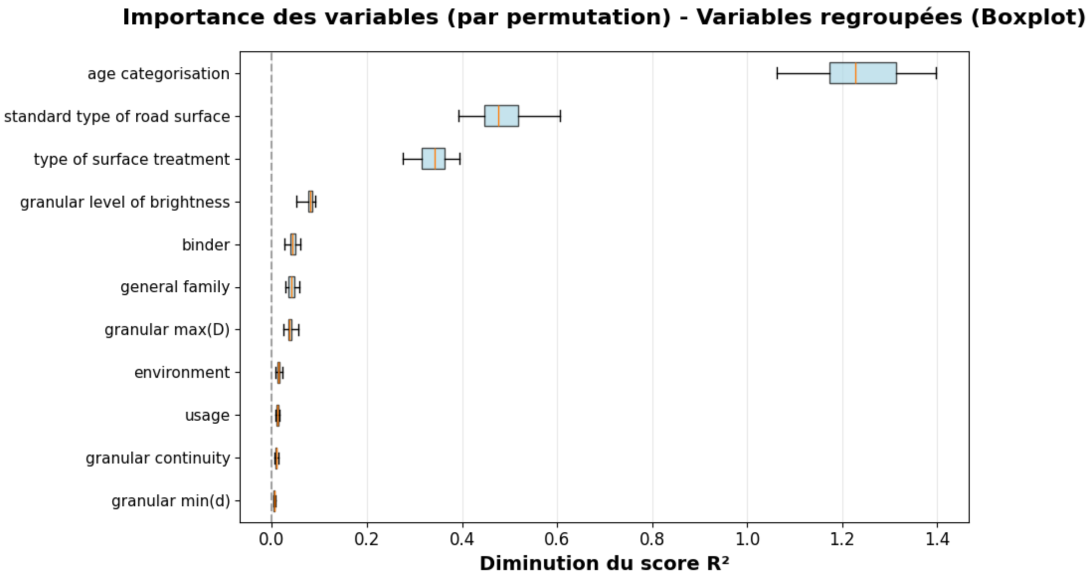


Figure 16: Permutation feature importance sorted by decreasing regression score (30 evaluations), Random Forest trained on roads younger than 24 months

Figure 16 shows that for roads younger than 24 months, age is still the most important variable. We also observe, as expected, that surface treatment gains in importance, since it is on the youngest surfaces that surface treatments are effective.

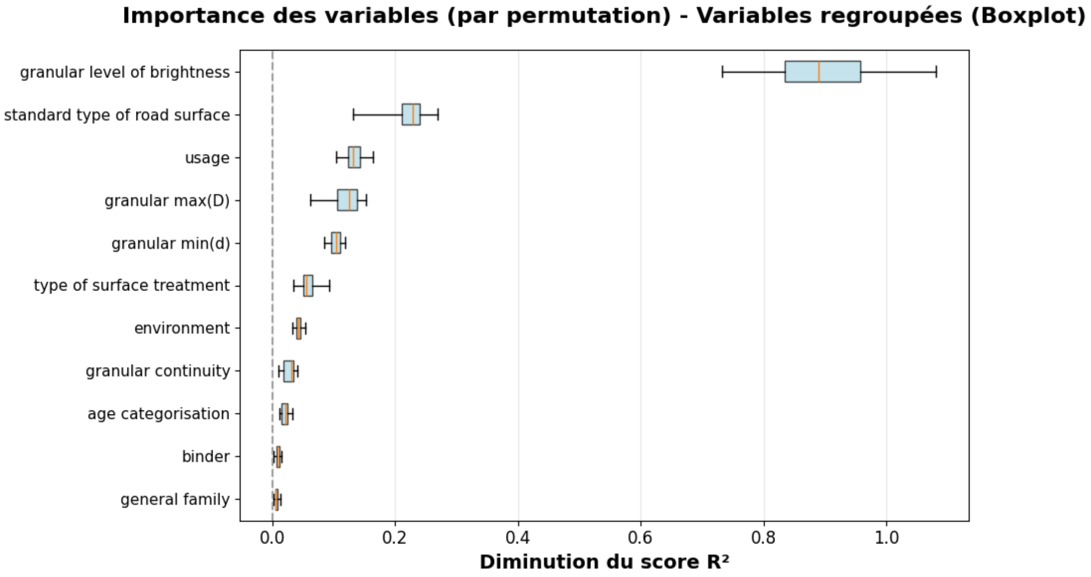


Figure 17: Permutation feature importance sorted by decreasing regression score (30 evaluations), Random Forest trained on road older than 24 months

Figure 17 illustrates the importance of each feature for the regression score of the Random

Forest model trained on roads older than 24 months. It shows, as expected, that the age of the road is no longer an important factor for the model. Moreover, the order of importance of variables has changed, which seems to indicate (according to the model) that the factors influencing road specularity are not the same for young roads and older roads. For example, young roads still have their binder layer that covers their grannular materials, so the importance of granular level of brightness only appears for aged roads, which is exactly what we verify here.

It should be kept in mind, however, that the model for older roads is significantly less effective than the one for younger roads, so these results should be interpreted with caution.

VII.2 Shap Values

VII.2.1 Theory

Shapley values are based on game theory and serve to fairly divide the rewards of a game among the participants. The Shapley value is the expected marginal contribution of each player. It is calculated by taking the average of the contributions that a player would bring to each coalition they can join. [20] [21]

Equation 6 gives the formula for the Shapley value of player i of a p player game. Starting with the summation sign, we are summing over all coalitions S. Where S is the subset of coalitions that do not include Player i. In other words, S contains all the coalitions to which Player i is able to make a marginal contribution.

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(p - |S| - 1)!}{p!} [v(S \cup \{i\}) - v(S)] \tag{6}$$

- $N = \{1, ..., p\}$ is the set of all players in the game.
- $[v(S \cup \{i\}) - v(S)]$ represents the marginal contribution of player i to the coalition S. It measures how much the value of the coalition increases when player i joins it.
- $\frac{|S|!(p-|S|-1)!}{p!}$ represents the weight of the coalition S. It is the number of ways to choose a coalition S of size $|S|$ from p players, multiplied by the number of ways to choose the remaining players from the remaining $(p - |S| - 1)$ players, divided by the number of ways to form a coalition of p players $p!$. This weight ensures that each coalition is counted equally in the calculation of the Shapley value.

The Shapley value verifie 3 axioms. These axioms can be considered a definition of fairness. Hence, a method of dividing value that satisfies this definition can be considered fair.

These axioms are:

- Efficiency : The sum of the Shapley values of all agents equals the value of the grand coalition, so that all the gain is distributed among the agents.

  $\sum_{i \in N} \phi_i(v) = v(N)$

- Symmetry : Two players are considered interchangeable if they make the same contributions to all coalitions. If two players are interchangeable then they must be given an equal share of the game's total value.

  $v(S \cup \{i\}) = v(S \cup \{j\}) \Rightarrow \varphi_i(v) = \varphi_j(v)$

- Null player property : If a player makes zero marginal contribution to all coalitions then they get none of the total value.

- Additivity / Linearity : If we combine two games, then a player's overall contribution is the sum of the contributions for the two individual games. This axiom makes the assumption that any games played are independent.

  $\varphi_i(v + w) = \varphi_i(v) + \varphi_i(w)$

Shapley (who gives his name to Shapley values) proved that the values calculated with Equation 6 are the only ones to satisfy these 4 axioms. [20]

VII.2.2 Applying SHAP values to our regression analysis

We can use Shapley values to understand how a model has made a prediction. Now, the value of the game is the model prediction. The feature values are the players. To be clear, it is not the features but the values of the features for a particular observation that play the game. However, we will refer to these as the features. We use Shapley values to calculate how each of the features has contributed to the prediction. [22] [23] [21]

The Python library SHAP [24] implements SHAP values for machine learning models. It enables efficient computation of SHAP values and visualization of their impact on model predictions. For further details on how the SHAP library computes SHAP values, see [21] [22] [23].

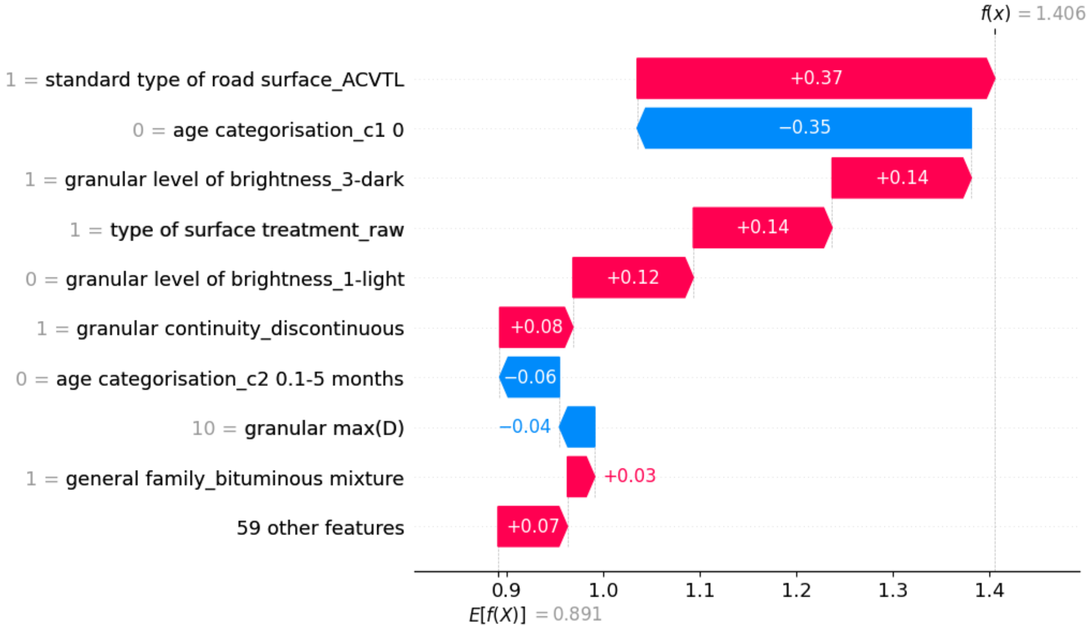


Figure 18: Visualisation of SHAP values for a single prediction

You can see the Shapley values for a particular observation in Figure 18. These give the difference between the average predicted age, $E[f(x)]$ (at the bottom of the figure), and the observation's prediction, $f(x)$ (on top). For example, the value for "granular level of brightness_dark" is 1, which means that thegranular level of brightness of the particular sample is dark, and this causes the model to increase the predicted specularity by 0.14.

It is readily apparent that one-hot encoding of features, which increases the number of features, can create confusion in understanding the impact of these features. To address this, methods have been developed to visualize the impact of features along with their induced impact. For example, by definition of one-hot encoding, if `granular level of brightness_dark` = 1, then all other features `standard type of road surface_XXX` are forced to 0. By summing the `shap_values` for `granular level of brightness_dark` = 1 and the `shap_values` for all `standard type of road surface_XXX` = 0, the true impact of `granular level of brightness_dark` = 1 is obtained. This adjustment allows for clearer representations, as shown in Figure 19, which displays the SHAP values with induced impact (for the same input as in Figure 18).

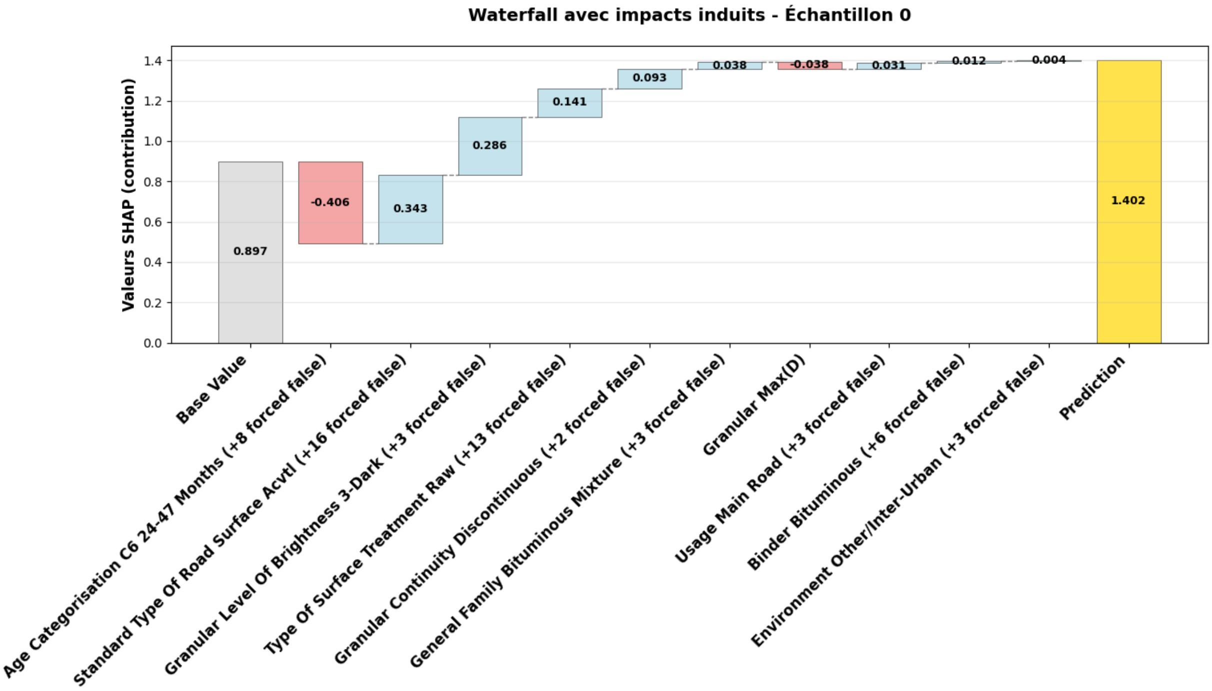


Figure 19: Visualisation of SHAP values for a single prediction with induced impact, base value $= E[f(x)]$, prediction $= f(x)$

To study the average impact of each feature, these SHAP values with induced impact are used and then averaged over all observations. This approach provides the mean impact of each feature. For example, Figure 20 shows that the feature `granular level of brightness_dark = 1` increases the specularity by 0.123 on average.

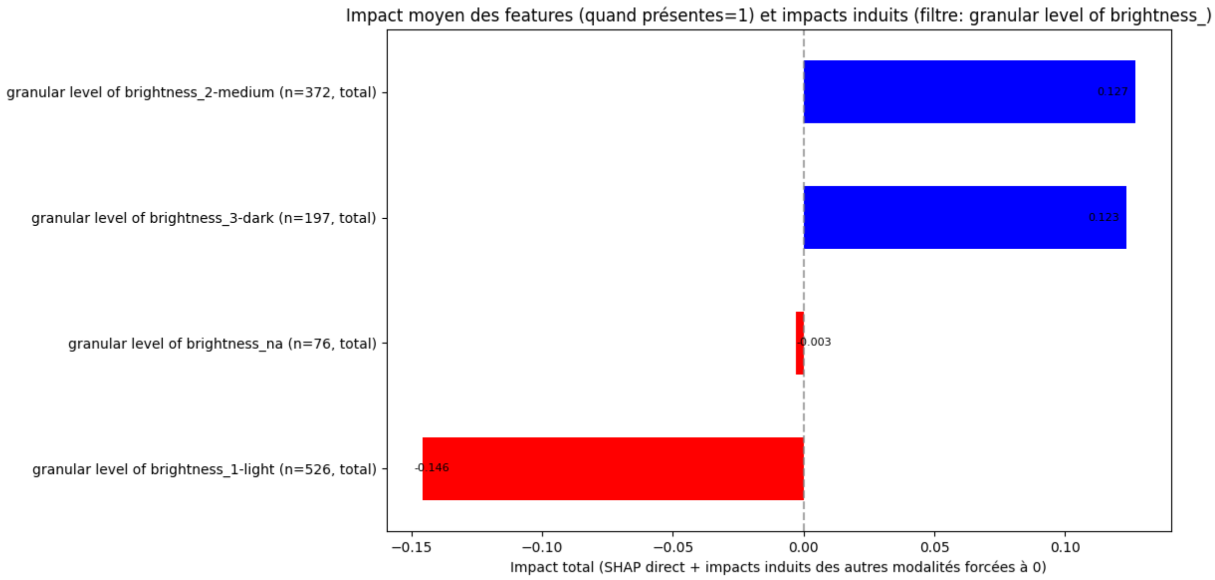


Figure 20: Visualisation of SHAP values related to granular level of brightness with induced impact, averaged over all observations

The complete figure of all SHAP values with induced impact is available in Appendix section V. It highlights, for example, the strong impact that a new condition has on high specularity, while initial treatments systematically result in a decrease in specularity.

VII.3 Construction of a model using mean SHAP values with induced impact

With the mean SHAP values with induced impact, it is possible to construct a linear model of the form Equation 7, which eliminates the need for a complex model.

$$f(x) = E[f(x)] + \sum_{i=1}^{M} \phi_i \alpha_i \tag{7}$$

Here, $E[f(x)]$ is the mean value of specularity, ϕ_i are the mean SHAP values with induced impact for each feature, α_i is 0 if the feature is 0 and 1 if the feature is 1, and M is the number of features. This model is very simple to understand, interpret, and use by non-experts. It requires only the list of ϕ_i and the base value $E[f(x)]$.

This model provides results that are significantly less accurate than those of complex models, but it is not without interest, as it is very easy to use. In Table 17, a comparison of the performance between the optimized RandomForest and the model based on mean SHAP values with induced impact is presented; both models are evaluated on the entire dataset.

Metric	**Optimised RandomForest**	**Custom linear Model**
r^2	0,823	0,475
Accuracy (3 classes)	0,85	0,74

Table 17: Comparison of performances between Optimised RandomForest and my SHAP values based model.

On Figure 21, the low coefficient of determination of the model based on SHAP values is visually confirmed, particularly for high specularity values. The model appears to have a ceiling, which is likely due to its linear nature. Indeed, very high specularity values are often the result of interactions and accumulation of factors that combine in a non-linear manner. For example, the combination of a low age and the absence of surface treatment. It is also observed that increasing the number of classes is likely to quickly deteriorate the model's accuracy.

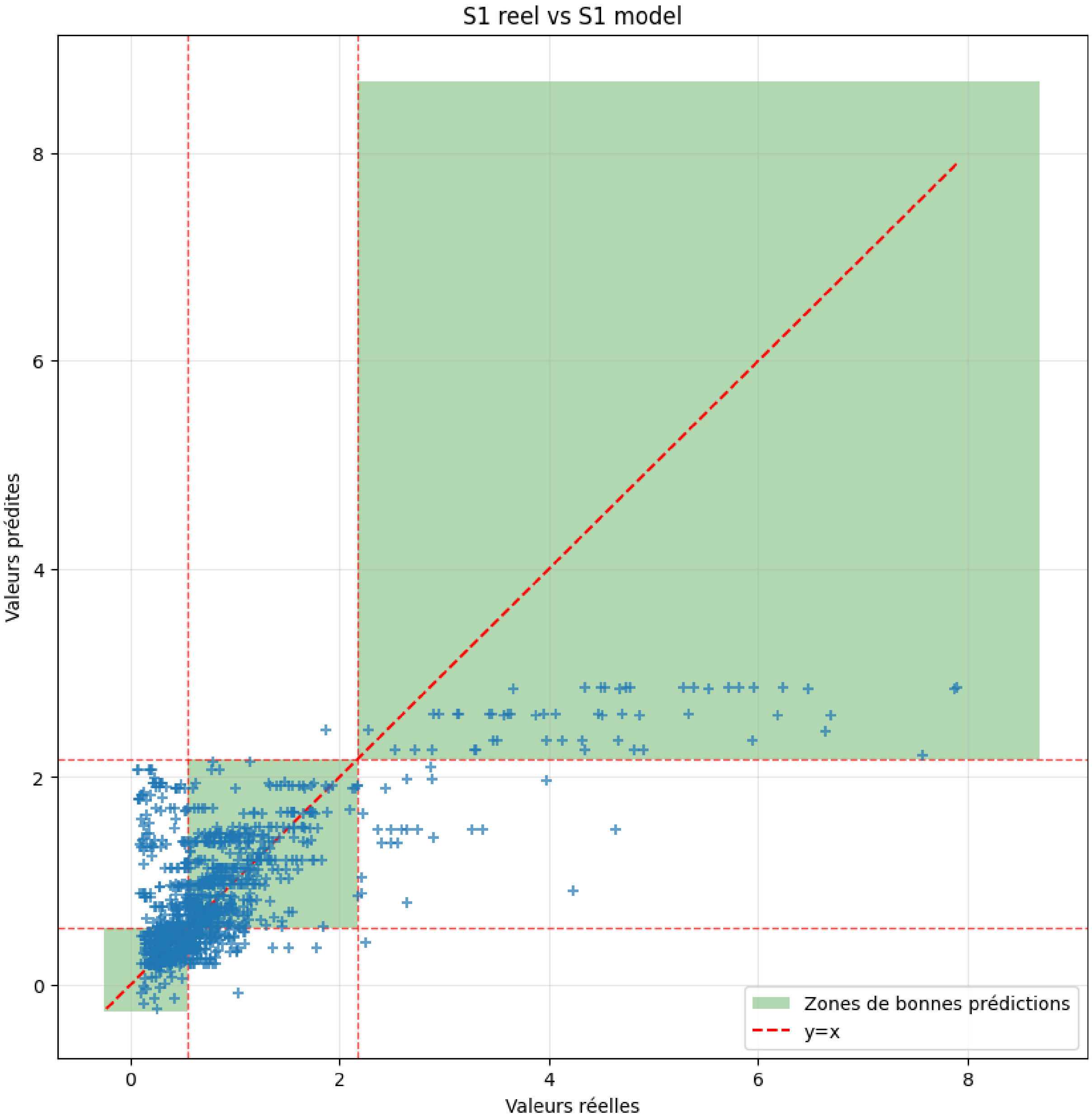


Figure 21: Visualisation of the SHAP_values_based model predictions

In conclusion, this attempt to avoid a complex model appears to be less effective, but it may be of interest for classification tasks with a small number of classes as shown in Table 17 and Figure 21.

VIII Work environment and code transmission

To make the work carried out during this internship accessible, the code was designed to be easily usable by others. In particular, this was achieved through easily executable notebooks, which allow analyses to be reproduced and data to be explored interactively.

The environment, including a Python environment with the necessary libraries, the required software for using MongoDB, and a Linux virtual machine to run everything, was configured and containerized using Docker and VS Code's containerization tools. This setup enables easy installation and ensures consistent operation across different machines.

All the code is available on CEREMA's GitLab but is not publicly accessible. The code has been shared with the project's partners.

IX Conclusion and perspectives

To optimize road lighting, knowledge of the photometric properties of road surfaces is essential. Since measurements are rarely performed in practice, developing a prediction model based on material properties and surface age would be highly valuable for lighting designers. To achieve this goal, an international data collection was carried out within the framework of CIE TC4-50 and using the Cerema COLUROUTE measurement device. The dataset includes Excel files containing metadata describing the road surface (composition, location, age, etc.) and the measured photometric data.

In summary, this work established a structured and standardized MongoDB database from heterogeneous Excel sources, enabling comprehensive data cleaning, formatting, and the creation of new standardized features. Visualization techniques facilitated preliminary and graphical analyses, providing valuable insights into the dataset. In particular, the results were consistent with the literature (effect of surface age, effect of using lighter aggregates, etc.), both for the specularity factor S1 and the average luminance coefficient Q0. As the CIE proposes a classification of road surface reflection properties based on the specularity indicator, this indicator was a central focus in our modeling work.

Several machine learning models were tested to predict the specularity S1 from the standardized features selected after exploratory data analysis. The model giving the best performance was the random forest, with a coefficient of determination of 0.823 and an accuracy of 0.85 in the best scenario. We also implemented feature importance interpretation methods based on SHAP indicators and a permutation approach to identify the key factors influencing surface reflectivity.

Finally, a linear model based on these interpretability results was proposed as a simplified alternative to more complex approaches. In addition to S1 prediction performance, the classification performance of the models was compared using predefined thresholds.

All developed code is available on the Cerema platform and is accompanied by Jupyter notebooks to facilitate its use, particularly within the context of CIE TC4-50 activities.

The findings of this work contribute to a better understanding of road surface behavior and can inform future research and practical applications in the field of road infrastructure and lighting design. The developed method is highly modular: it can be applied to other datasets if the database is expanded. For S1-based classification, the thresholds are easily configurable and can be adapted to future CIE recommendations. Finally, the entire approach can readily be extended to other global indicators, such as the average luminance coefficient Q0.

Bibliography

[1] V. Muzet, F. Greffier, and H. Mrad. Review on factors influencing the light reflection properties of road surfaces – part 1: Internal factors. *Lighting Research & Technology*, 2025. submitted.

[2] V. Muzet, F. Greffier, and H. Mrad. Review on factors influencing the light reflection properties of road surfaces – part 2: External factors. *Lighting Research & Technology*, 2025. submitted.

[3] reflectivity cerema. https://www.anr-reflectivity.fr/, 2025. Accessed: 2025-05-27.

[4] CIE. https://cie.co.at/technicalcommittees/road-surface-characterization-lighting-applications.

[5] Valerie Muzet, Jean-Luc Paumier, and Yannick Guillard. COLUROUTE : a mobile gonio-reflectometer to characterize the road surface photometry. In *The 2nd CIE expert symposium on "Advances in photometry and colorimetry"*, volume CIE x033:2008, Turin, ITALY, July 2008. CIE.

[6] V. Muzet, J.-P. Christory, P. Gandon-Leger, J. Dherbecourt, J. Abdo, S. Liandrat, L. Monfront, and A. Nicolaï. Démarche originale du groupe de travail revêtements & lumière pour optimiser les projets d'éclairage public. *RGRA*, pages 50–60, 2020.

[7] J.-L. Paumier, G. Legouaix, P. Dupont, F. Aubert, and E. Dumont. *Propriétés photométriques des revêtements de chaussée.* CFTR, Paris, 2006.

[8] International Commission on Illumination (CIE). Road Surface and Road Marking Reflection Characteristics. Technical Report CIE 144:2001, CIE, Vienna, 2001.

[9] Vincent Boucher, Valérie Muzet, and Paola Iacomussi. Mathematical considerations for road reflection properties. In *Proceedings of the 30th Session of the CIE*, Ljubljana, Slovenia, September 2023. CIE.

[10] Valérie Muzet, Odile Balcer, and Aude Stresser. On site characterisation of road surfaces reflection properties for several observation angles. In *Proceedings of the CIE Midterm meeting*, Vienna, Austria, July 2025. CIE.

[11] Eric Dumont, Jean Luc Paumier, and Vincent Ledoux. Are standard r-tables still representative of road surface photometric characteristics in France? In *CIE International Symposium on Road Surface Photometric Characteristics*, page 8p, France, July 2008.

[12] V. Muzet, F. Greffier, and H. Mrad. Review on Factors Influencing the Light Reflection Properties of Road Surfaces – Part 1: Internal Factors. *Lighting Research & Technology*, 2025.

[13] Florian Greffier, Valérie Muzet, Vincent Boucher, Fabrice Fournela, Laure Lebouc, and Sébastien Liandrat. Influence of Pavement Heterogeneity and Observation Angle on Lighting Design: Study with New Metrics. *Sustainability*, 13(21):11789, 2021.

[14] MongoDB. MongoDB: The World's Leading Modern Database.

[15] MongoDB. pymongo: PyMongo - the Official MongoDB Python driver.

[16] Kai Sørensen. Road surface reflection data. Technical Report Report No. 10, The Danish Illuminating Engineering Laboratory, January 1975.

[17] V. Muzet, F. Greffier, and H. Mrad. Review on Factors Influencing the Light Reflection Properties of Road Surfaces – Part 2: External Factors. *Lighting Research & Technology*, 2025.

[18] Meslin Hillion. Introduction aux méthodes ensemblistes. Technical report, Insee, 2025.

[19] Scikit-Learn. Scikit-Learn API Reference https://scikit-learn.org/stable/api/index.html.

[20] Lloyd S Shapley et al. A value for n-person games. *torrossa*, 1953.

[21] Conor O'Sullivan. From Shapley to SHAP — Understanding the Math, March 2023.

[22] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. *Nature Machine Intelligence*, 2017.

[23] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):56–67, January 2020. Publisher: Nature Publishing Group.

[24] SHAP documentation — https://shap.readthedocs.io/en/latest/index.html.

Appendix

I New features graphic

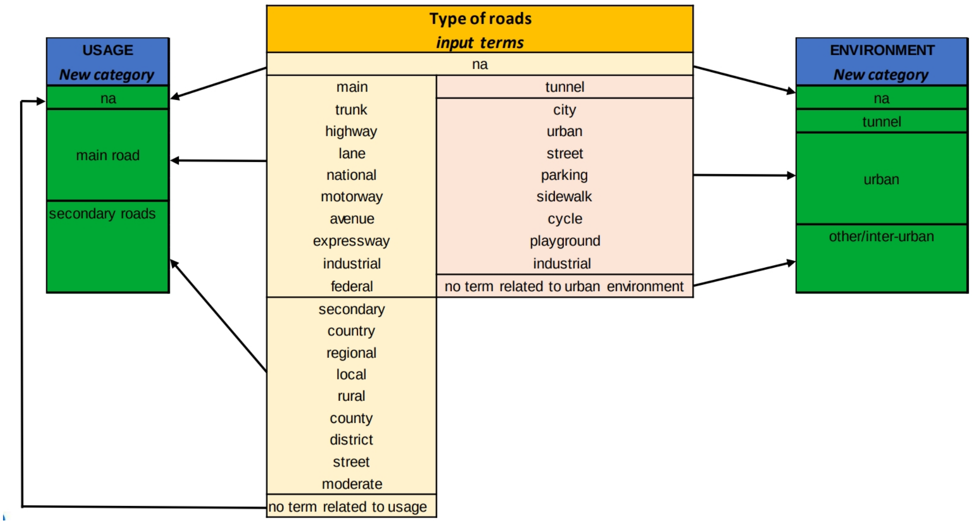


Figure 22: Creation of features environment and usage

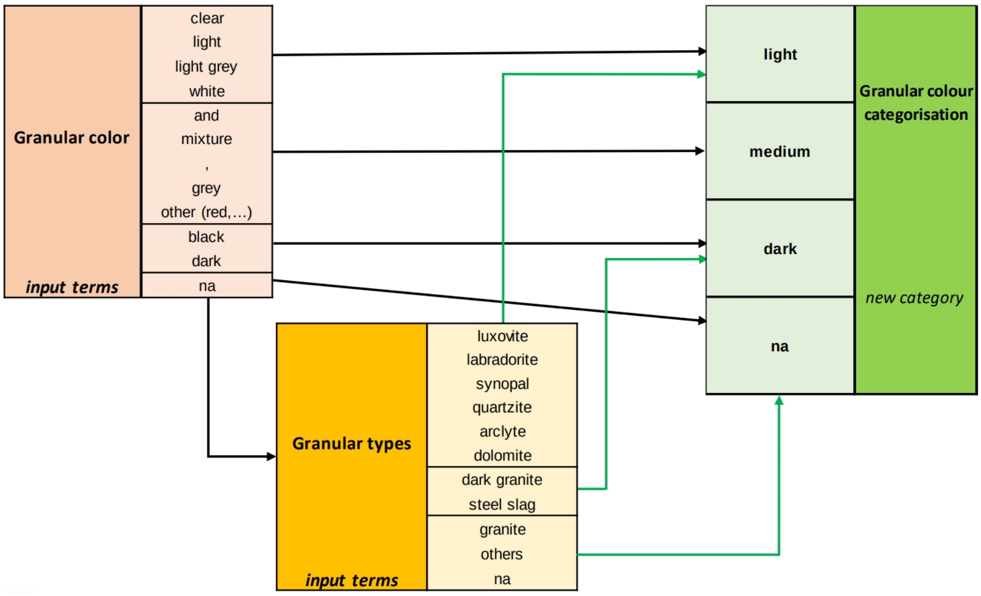
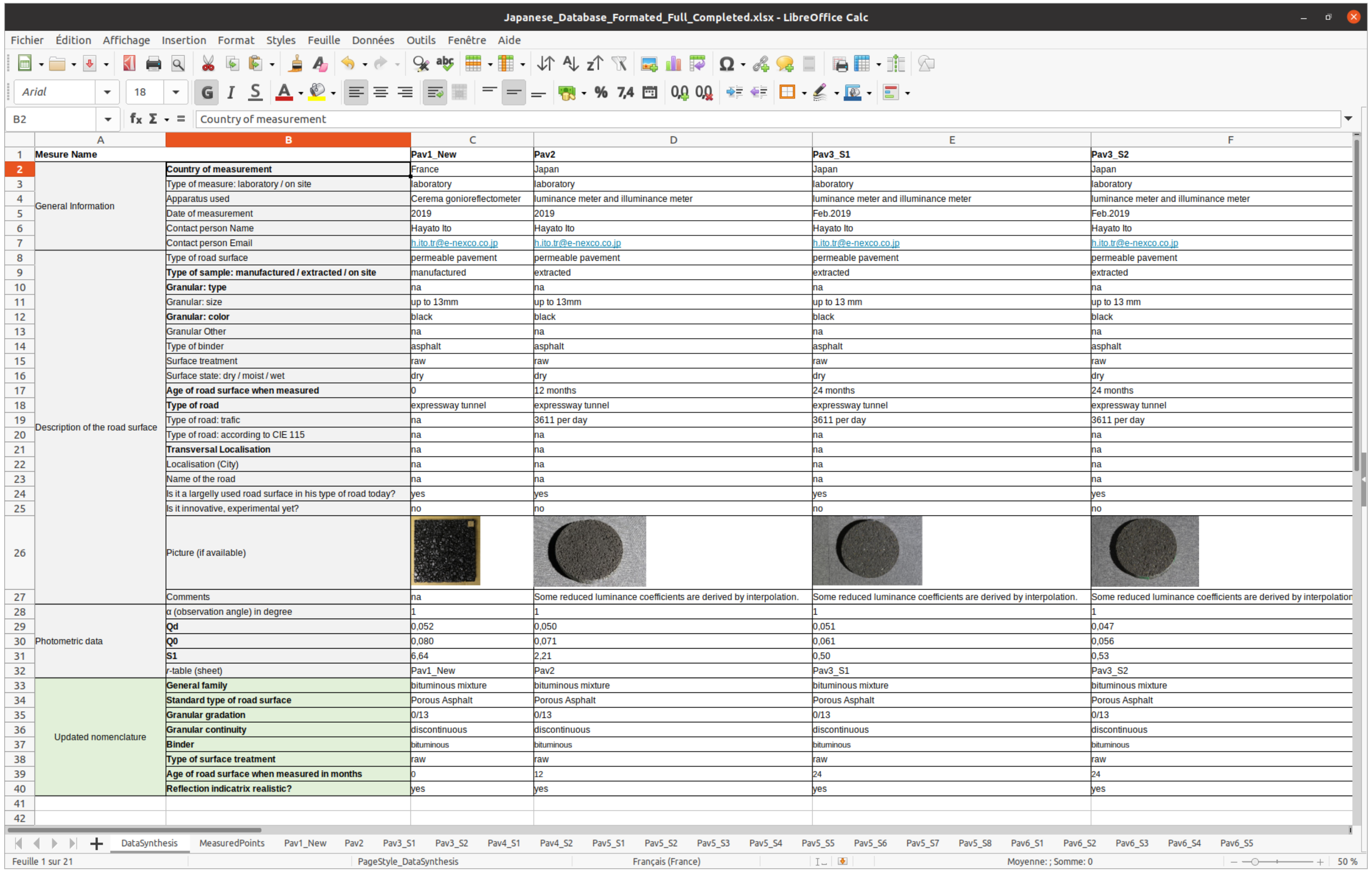


Figure 23: Creation of features color categorization

II Dataset example


Japanese_Database_Formated_Full_Completed.xlsx - LibreOffice Calc
Fichier Édition Affichage Insertion Format Styles Feuille Données Outils Fenêtre Aide
Arial
18
B2
Country of measurement
DataSynthesis MeasuredPoints Pav1_New Pav2 Pav3_S1 Pav3_S2 Pav4_S1 Pav4_S2 Pav5_S1 Pav5_S2 Pav5_S3 Pav5_S4 Pav5_S5 Pav5_S6 Pav5_S7 Pav5_S8 Pav6_S1 Pav6_S2 Pav6_S3 Pav6_S4 Pav6_S5
Feuille 1 sur 21
PageStyle_DataSynthesis
Français (France)
Moyenne: ; Somme: 0
50 %

	A	B	C	D	E	F
1	**Mesure Name**		**Pav1_New**	**Pav2**	**Pav3_S1**	**Pav3_S2**
2	General Information	**Country of measurement**	France	Japan	Japan	Japan
3		Type of measure: laboratory / on site	laboratory	laboratory	laboratory	laboratory
4		Apparatus used	Cerema gonioreflectometer	luminance meter and illuminance meter	luminance meter and illuminance meter	luminance meter and illuminance meter
5		Date of measurement	2019	2019	Feb.2019	Feb.2019
6		Contact person Name	Hayato Ito	Hayato Ito	Hayato Ito	Hayato Ito
7		Contact person Email	h.ito.tr@e-nexco.co.jp	h.ito.tr@e-nexco.co.jp	h.ito.tr@e-nexco.co.jp	h.ito.tr@e-nexco.co.jp
8	Description of the road surface	Type of road surface	permeable pavement	permeable pavement	permeable pavement	permeable pavement
9		**Type of sample: manufactured / extracted / on site**	manufactured	extracted	extracted	extracted
10		**Granular: type**	na	na	na	na
11		Granular: size	up to 13mm	up to 13mm	up to 13 mm	up to 13 mm
12		**Granular: color**	black	black	black	black
13		Granular Other	na	na	na	na
14		Type of binder	asphalt	asphalt	asphalt	asphalt
15		Surface treatment	raw	raw	raw	raw
16		Surface state: dry / moist / wet	dry	dry	dry	dry
17		**Age of road surface when measured**	0	12 months	24 months	24 months
18		**Type of road**	expressway tunnel	expressway tunnel	expressway tunnel	expressway tunnel
19		Type of road: trafic	na	3611 per day	3611 per day	3611 per day
20		Type of road: according to CIE 115	na	na	na	na
21		**Transversal Localisation**	na	na	na	na
22		Localisation (City)	na	na	na	na
23		Name of the road	na	na	na	na
24		Is it a largelly used road surface in his type of road today?	yes	yes	yes	yes
25		Is it innovative, experimental yet?	no	no	no	no
26		Picture (if available)				
27		Comments	na	Some reduced luminance coefficients are derived by interpolation.	Some reduced luminance coefficients are derived by interpolation.	Some reduced luminance coefficients are derived by interpolation
28	Photometric data	α (observation angle) in degree	1	1	1	1
29		**Qd**	0,052	0,050	0,051	0,047
30		**Q0**	0,080	0,071	0,061	0,056
31		**S1**	6,64	2,21	0,50	0,53
32		r-table (sheet)	Pav1_New	Pav2	Pav3_S1	Pav3_S2
33	Updated nomenclature	**General family**	bituminous mixture	bituminous mixture	bituminous mixture	bituminous mixture
34		**Standard type of road surface**	Porous Asphalt	Porous Asphalt	Porous Asphalt	Porous Asphalt
35		**Granular gradation**	0/13	0/13	0/13	0/13
36		**Granular continuity**	discontinuous	discontinuous	discontinuous	discontinuous
37		**Binder**	bituminous	bituminous	bituminous	bituminous
38		**Type of surface treatment**	raw	raw	raw	raw
39		**Age of road surface when measured in months**	0	12	24	24
40		**Reflection indicatrix realistic?**	yes	yes	yes	yes
41						
42						

Figure 24: dataset example

III Extract of keys, values and repartition

key	valeur	count	realistic	pourcentage (count/ total)
type of sample: manufactured / extracted / on site	extracted	670	526	78.36
type of sample: manufactured / extracted / on site	manufactured	138	60	16.14
type of sample: manufactured / extracted / on site	na	41	37	4.80
type of sample: manufactured / extracted / on site	on site	6	6	0.70
granular : type	nan	1	1	0.12
granular : type	100% luxovite	1	1	0.12
granular : type	13% luxovite and 13% synopal	8	8	0.94
granular : type	15% luxovite	10	10	1.17
granular : type	15% luxovite and 57% granite	1	1	0.12
granular : type	15% luxovite and 85% dark granite	10	6	1.17
granular : type	15% luxovite and crushed mat.	1	1	0.12
granular : type	15% luxovite and granite	5	4	0.58
granular : type	15% luxovite and steel slag	1	1	0.12
granular : type	15% synopal	5	4	0.58
granular : type	15% synopal and 43% granite	3	3	0.35
granular : type	16% synopal and 16% hyperite	3	3	0.35
granular : type	17% luxovite and 22% gravel pit mat.	1	1	0.12
granular : type	17% luxovite and 35% hyperite	1	1	0.12
granular : type	17% luxovite and 47% dark granite	1	1	0.12
granular : type	17% luxovite and 55% quartzite	1	1	0.12
granular : type	19% synopal	1	1	0.12
granular : type	19% synopal and granite	1	1	0.12
granular : type	20% labradorite and granite	1	1	0.12
granular : type	20% luxovite and 50% hyperite	3	3	0.35
granular : type	20% synopal and 60% dark granite	1	1	0.12
granular : type	20% synopal and 80 % dark granite	1	1	0.12
granular : type	21% luxovite and 45% quartzite	1	1	0.12
granular : type	21% luxovite and 48% quartzite	1	1	0.12
granular : type	21% luxovite and 51% quartzite	4	4	0.47
granular : type	21% luxovite and 53% quartzite	5	5	0.58
granular : type	25% luxovite and 75% dark granite	1	0	0.12
granular : type	25% synopal	1	1	0.12
granular : type	25% synopal and 55% granite	1	1	0.12
granular : type	25% synopal and granite	1	1	0.12
granular : type	30% synopal and 70% granite	3	2	0.35
granular : type	31% quartzite	4	4	0.47
granular : type	35% luxovite and gravel pit mat.	9	9	1.05
granular : type	37% granite	6	4	0.70
granular : type	40% synopal and 60 % granite	7	6	0.82
granular : type	45% quartzite	4	4	0.47
granular : type	47% luxovite and 47% quartzite	4	4	0.47

key	valeur	count	realistic	pourcentage (count/ total)
granular : type	48% synopal and 15% quartzite	4	4	0.47
granular : type	50% luxovite and 50% quartzite	3	3	0.35
granular : type	52% quartzite	4	4	0.47
granular : type	62% quartzite	4	4	0.47
granular : type	aggregates alluvial	8	8	0.94
granular : type	arclyte	4	4	0.47
granular : type	basalt	10	1	1.17
granular : type	continuous	7	7	0.82
granular : type	crushed mat.	5	5	0.58
granular : type	crushed stone	74	0	8.65
granular : type	dark granite	20	15	2.34
granular : type	dolomite	1	1	0.12
granular : type	granite	10	10	1.17
granular : type	granite and crushed mat.	1	1	0.12
granular : type	granite and hyperite	2	1	0.23
granular : type	granite and labradorite	3	3	0.35
granular : type	gravel pit mat.	2	2	0.23
granular : type	labradorite	4	4	0.47
granular : type	light granites	1	1	0.12
granular : type	limestone	15	1	1.75
granular : type	loose rock, crushed	1	0	0.12
granular : type	luxovite	46	45	5.38
granular : type	massive and loose rock, crushed	9	5	1.05
granular : type	massive rock, crushed	29	11	3.39
granular : type	microgranit	41	41	4.80
granular : type	na	205	139	23.98
granular : type	porphyre	39	36	4.56
granular : type	quartzite	56	56	6.55
granular : type	rock, crushed	18	4	2.11
granular : type	sandstone quartzeux	41	41	4.80
granular : type	slag	4	2	0.47
granular : type	spilite	41	40	4.80
granular : type	steel slag	5	4	0.58
granular : type	steel stage, dark granite	1	1	0.12
granular : type	synopal	24	23	2.81
granular : color	black	60	58	7.02
granular : color	clear	33	28	3.86
granular : color	dark	14	11	1.64
granular : color	dark grey	29	2	3.39
granular : color	greenish-grey	1	1	0.12
granular : color	grey	143	70	16.73
granular : color	grey (mixture)	6	0	0.70
granular : color	grey and clear	1	0	0.12
granular : color	grey-green	42	41	4.91
granular : color	grey/light grey (mixture)	10	4	1.17

key	valeur	count	realistic	pourcentage (count/ total)
granular : color	grey/yellowish (mixture)	1	0	0.12
granular : color	light	64	36	7.49
granular : color	light grey	2	1	0.23
granular : color	mixture	7	2	0.82
granular : color	na	333	276	38.95
granular : color	normal	3	3	0.35
granular : color	not visible	5	0	0.58
granular : color	red	41	41	4.80
granular : color	reddish	3	0	0.35
granular : color	reddish/light grey (mixture)	1	1	0.12
granular : color	white	53	52	6.20
granular : color	yellowish	2	2	0.23
granular : color	yellowish / grey (mixture)	1	0	0.12
type of binder	asphalt	38	22	4.44
type of binder	asphalt 25/55-55 a	7	3	0.82
type of binder	asphalt 25/55-55 a + saso	2	2	0.23
type of binder	asphalt 25/55-55 a nv	1	0	0.12
type of binder	asphalt 25/55-55 a tr	1	0	0.12
type of binder	asphalt 45/80-65	1	0	0.12
type of binder	asphalt 50/70	3	0	0.35
type of binder	asphalt 70/100	3	1	0.35
type of binder	asphalt b30/70	1	1	0.12
type of binder	asphalt b65	5	0	0.58
type of binder	asphalt pmb 45 a	2	1	0.23
type of binder	bitumen	44	14	5.15
type of binder	bitumen (asphalt)	74	0	8.65
type of binder	bituminen	2	1	0.23
type of binder	bituminous	268	257	31.35
type of binder	cement	50	33	5.85
type of binder	cement?	1	1	0.12
type of binder	modified bitumen pmb 25- 55/75	4	2	0.47
type of binder	na	310	265	36.26
type of binder	reaction resin	7	6	0.82
type of binder	sbs modified asphalt cement	7	7	0.82
type of binder	synthetic	4	2	0.47
type of binder	synthetic, tio2 add	20	11	2.34
surface treatment	aged	2	1	0.23
surface treatment	broomed	1	1	0.12
surface treatment	brushed	4	4	0.47
surface treatment	bush hammered	4	4	0.47
surface treatment	completely covered by asphalt	4	0	0.47
surface treatment	deactivated	9	8	1.05
surface treatment	es	2	0	0.23
surface treatment	es?	1	1	0.12
surface treatment	flamed	12	3	1.40

key	valeur	count	realistic	pourcentage (count/ total)
surface treatment	na	316	266	36.96
surface treatment	no	17	10	1.99
surface treatment	raw	387	271	45.26
surface treatment	sand blasted	6	0	0.70
surface treatment	sand-blasted	10	6	1.17
surface treatment	shot blasted	6	5	0.70
surface treatment	smoothed	2	1	0.23
surface treatment	swiped	2	1	0.23
surface treatment	top asphalt layer removed by blasting with glass beads	17	9	1.99
surface treatment	top asphalt layer removed by sand blasted	5	0	0.58
surface treatment	top cement layer brushed out	2	0	0.23
surface treatment	top cement layer textured	3	1	0.35
surface treatment	washed	2	1	0.23
surface treatment	water jet scrubbing	16	13	1.87
surface treatment	yes	19	17	2.22
surface treatment	yes+	4	4	0.47
surface treatment	yes++	2	2	0.23
general family	bituminous mixture	731	537	85.50
general family	cement concrete	61	44	7.13
general family	natural material	8	1	0.94
general family	surface coating	55	47	6.43
standard type of road surface	ac	231	123	27.02
standard type of road surface	acbe	39	35	4.56
standard type of road surface	actl	32	28	3.74
standard type of road surface	acvtl	174	168	20.35
standard type of road surface	hot rolled asphalt	25	22	2.92
standard type of road surface	hot rolled asphalt or topeka	13	13	1.52
standard type of road surface	mastic asphalt	52	39	6.08
standard type of road surface	medium coarse asphalt	7	0	0.82
standard type of road surface	na	53	38	6.20
standard type of road surface	porous asphalt	27	17	3.16
standard type of road surface	poured cement concrete	47	35	5.50
standard type of road surface	precast cement concrete	15	10	1.75
standard type of road surface	precast natural stone	8	1	0.94
standard type of road surface	sd	48	41	5.61
standard type of road surface	sma	34	9	3.98
standard type of road surface	topeka	50	50	5.85
granular continuity	continuous	331	273	38.71
granular continuity	discontinuous	72	49	8.42
granular continuity	na	452	307	52.87
binder	bituminous	753	565	88.07
binder	cement	61	44	7.13
binder	na	8	1	0.94
binder	reaction resin	7	6	0.82
binder	synthetic, tio2 add	26	13	3.04

key	valeur	count	realistic	pourcentage (count/ total)
type of surface treatment	aged	2	1	0.23
type of surface treatment	brushed	18	12	2.11
type of surface treatment	bush hammered	4	4	0.47
type of surface treatment	coated chipping	122	116	14.27
type of surface treatment	deactivated	9	8	1.05
type of surface treatment	flamed	12	3	1.40
type of surface treatment	na	32	3	3.74
type of surface treatment	raw	571	426	66.78
type of surface treatment	sand blasted	21	6	2.46
type of surface treatment	shot blasted	6	5	0.70
type of surface treatment	shot blasting	16	8	1.87
type of surface treatment	textured	1	1	0.12
type of surface treatment	washed	2	1	0.23
type of surface treatment	water jet scrubbing	16	13	1.87
type of surface treatment	yes	23	22	2.69
age of road surface when measured in months	0	140	87	16.37
age of road surface when measured in months	1	3	2	0.35
age of road surface when measured in months	2	2	2	0.23
age of road surface when measured in months	3	26	26	3.04
age of road surface when measured in months	6	26	26	3.04
age of road surface when measured in months	8	10	9	1.17
age of road surface when measured in months	9	1	1	0.12
age of road surface when measured in months	12	75	70	8.77
age of road surface when measured in months	14	1	1	0.12
age of road surface when measured in months	18	24	24	2.81
age of road surface when measured in months	23	9	7	1.05
age of road surface when measured in months	24	74	71	8.65
age of road surface when measured in months	30	48	23	5.61
age of road surface when measured in months	36	92	81	10.76
age of road surface when measured in months	42	5	5	0.58
age of road surface when measured in months	48	34	33	3.98
age of road surface when measured in months	54	4	2	0.47
age of road surface when measured in months	60	30	28	3.51
age of road surface when measured in months	72	24	17	2.81
age of road surface when measured in months	84	20	18	2.34
age of road surface when measured in months	96	9	8	1.05
age of road surface when measured in months	108	6	5	0.70
age of road surface when measured in months	120	8	8	0.94
age of road surface when measured in months	132	7	7	0.82
age of road surface when measured in months	144	2	2	0.23
age of road surface when measured in months	168	2	2	0.23
age of road surface when measured in months	180	2	2	0.23
age of road surface when measured in months	252	3	3	0.35
age of road surface when measured in months	300	1	1	0.12
age of road surface when measured in months	2	11	10	1.29

key	valeur	count	realistic	pourcentage (count/ total)
age of road surface when measured in months	24	75	47	8.77
age of road surface when measured in months	99	81	1	9.47
reflection indicatrix realistic?	no	226	629	26.43
reflection indicatrix realistic?	yes	629	629	73.57

IV Visual analysis

Effect of the type of measurement and sample

To analyse this effet, the used feature is “type of sample” because all in site measurements where conducted with COLUROUTE device and all the other types (manufactered, extracted and na) come from measurements conducted with laboratory gonioreflectometers (TC4-50 database). Figure 25 shows that manufactured samples have lower S1 and higher Q0 than the others. Moreover there does not seem to be a major difference between extracted samples measured with gonioreflectometers and on site measurements. So we will use all data for the factor analysis.

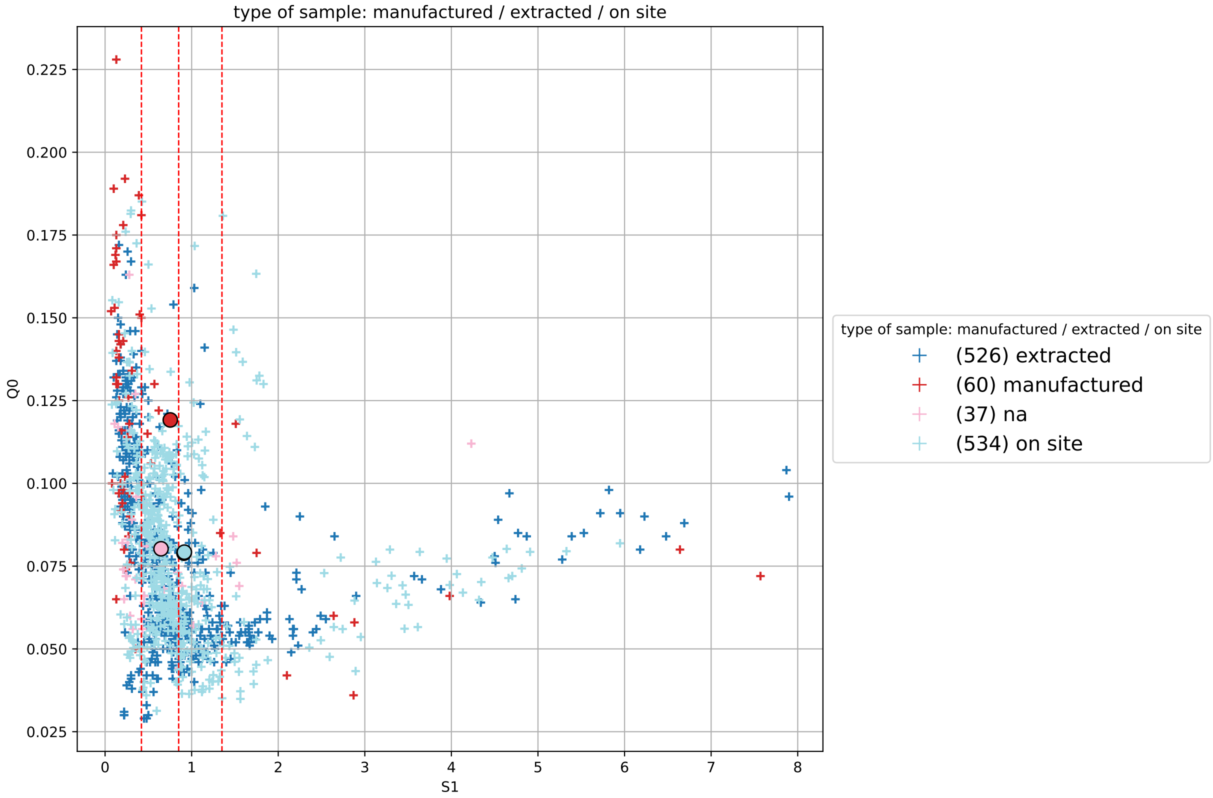


Figure 25: Effect of the type of sample on specularity and Q0

Binder effect

Figure 26 shows that cement concrete has lower specularity and higher Q0 than the classical bituminous binder. The use of synthetic binder (with TiO2 adds) instead of bituminous binder decreases the specularity and seriously increases Q0. These results are in accordance with the litterature [12].

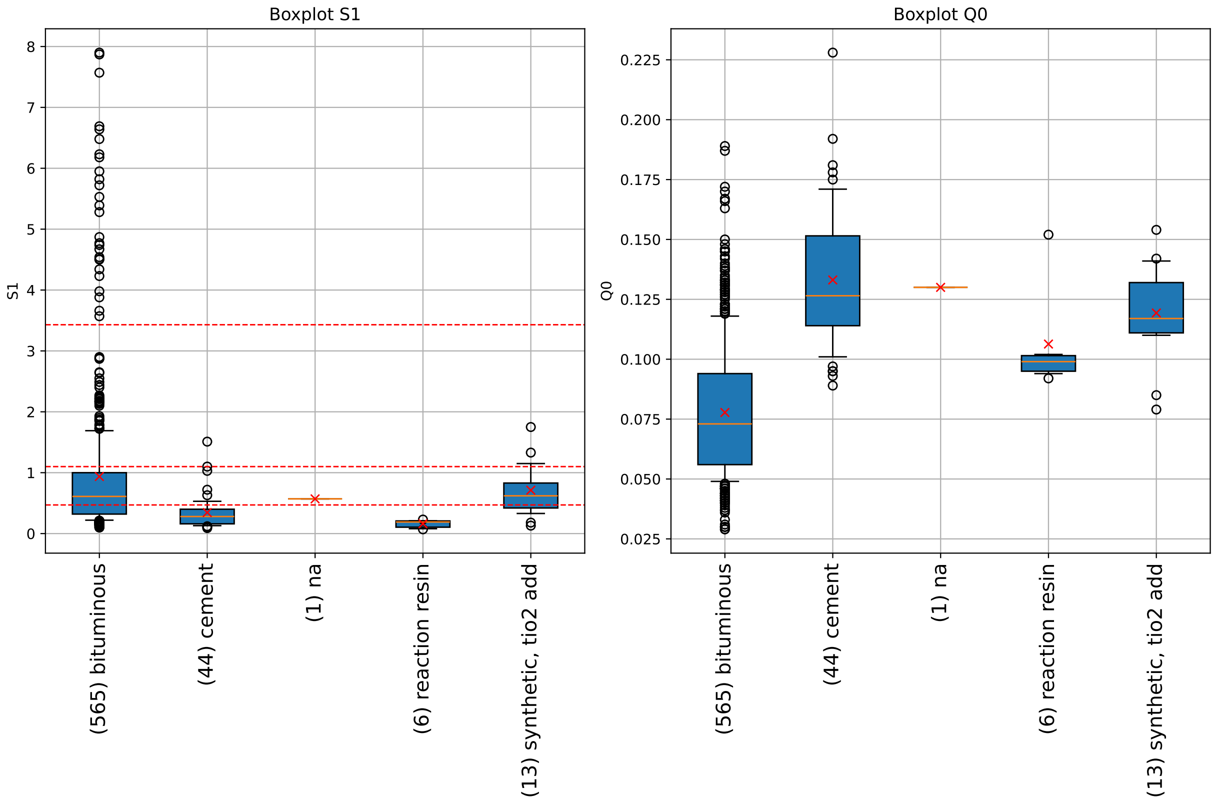


Figure 26: Effect of the binder type on specularity and Q0

V SHAP Values

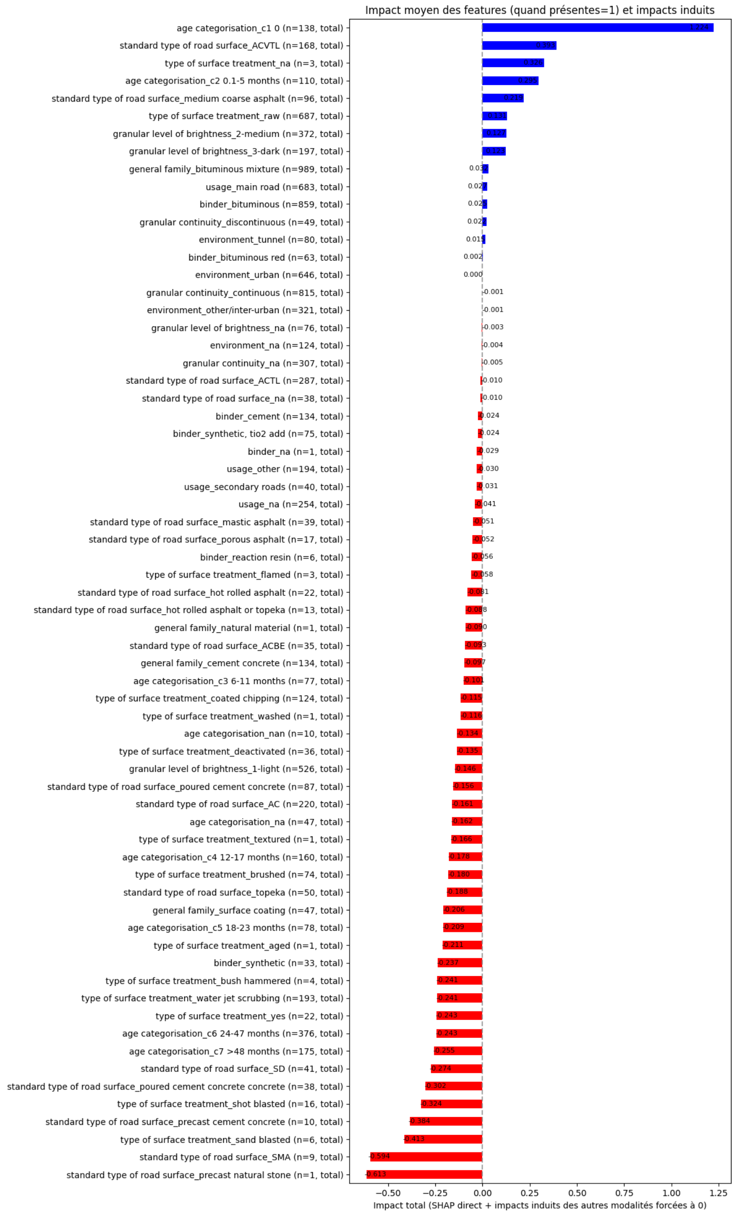


Figure 27: Visualisation of SHAP values with induced impact, averaged over all observations

VI Code user guide

User Guide for the DatabaseManager Class

General Overview

DatabaseManager is the main class for managing interactions with a MongoDB database. It allows you to insert, retrieve, clean, and analyze road reflectivity data.

Class Methods

`__init__(client, collection_name, db_name)`

- **Args:**
  - `client` (MongoClient): MongoDB client for database connection.
  - `collection_name` (str): Name of the main collection.
  - `db_name` (str): Name of the database.
- **Description:** Establishes the connection to the MongoDB database and sets up the necessary attributes for further operations.

`get_collection()`

- **Returns:** The active MongoDB collection.
- **Description:** Accessor method to get the reference to the active collection, allowing direct MongoDB operations if needed.

`set_collection(collection_name)`

- **Args:**
  - `collection_name` (str): Name of the new collection to use.
- **Description:** Updates the active collection within the same database. Useful for switching between collections without creating a new DatabaseManager instance.

`shut_client()`

- **Description:** Properly closes the MongoDB client connection to free resources. Important to avoid connection leaks, especially in long-running applications.

`insert_data(data_dir, reset=False, format=True)`

- **Args:**
  - `data_dir` (str): Path to the directory containing Excel files.
  - `reset` (bool): If True, empties the collections before inserting new data.
  - `format` (bool): If True, applies data formatting before insertion.
- **Description:** Scans all Excel files in the specified directory. For each file, the first sheet is considered the main data and is inserted into the main collection. Other sheets are considered r tables and are inserted into a dedicated collection. The `database` field is automatically added from the parent folder name.

`__clean_value(v, key=None)`

- **Args:**
  - `v`: Value to clean.
  - `key` (str, optional): Key associated with the value, to apply specific rules.
- **Returns:** The cleaned and standardized value.
- **Description:** Private function used for data cleaning. Applies various transformations depending on the data type and key (handling special values, string normalization, technical corrections, code and abbreviation standardization).

`__format_data(data)`

- **Args:**
  - `data` (list): List of dictionaries representing the records to format.
- **Returns:** List of formatted records.
- **Description:** Private function used before database insertion. Cleans each field, adds extra fields (environment, usage, age categorization, aggregate color, etc.).

`clean_mongo()`

- **Description:** Iterates through all documents in the active collection, applies cleaning rules to each field, and updates the documents. Useful for standardizing an existing database without recreating it.

`get_data(query=None, key=None, r_table=False)`

- **Args:**
  - `query` (dict, optional): MongoDB filter dictionary.
  - `key` (str or list, optional): Field or list of fields to include in the result.
  - `r_table` (bool): If True, also returns associated r tables.
- **Returns:** List of documents or tuple (documents, r tables).
- **Description:** Performs a query on the main collection and retrieves matching documents. If `r_table=True`, also searches for associated r tables in the `table_r` collection and returns them in the same order as the documents. If `key` is None, all fields are retrieved.

`write_key(which='all')`

- **Args:**
  - `which` (str): 'all' for all keys, 'usefull' for a predefined list.
- **Description:** Generates a CSV file containing, for each key-value combination, the key name, unique value, occurrence count, count of "realistic" documents with this value, and the percentage of occurrences. Helps analyze value distributions in the database.

`collection_stats()`

- **Description:** Computes and displays various statistics about the active collection: total number of entries, database size, data type distribution per field, global proportion, number of unique fields.

`benchmark_queries(queries, keys, n_repeat=10)`

- **Args:**
  - `queries` (list): List of dictionaries representing MongoDB queries.
  - `keys` (dict): Field projection to return.

User Guide for the `DatabaseManager` Class

General Overview

exttttDatabaseManager est la classe principale pour gérer les interactions avec une base MongoDB. Elle permet d'insérer, récupérer, nettoyer et analyser des données de réflectivité routière.

Class Methods

`__init__(client, collection_name, db_name)`

- **Args:**
  - `client` (MongoClient): Client MongoDB pour la connexion à la base.
  - `collection_name` (str): Nom de la collection principale.
  - `db_name` (str): Nom de la base de données.
- **Returns:** None
- **Description:** Établit la connexion avec la base de données MongoDB et configure les attributs nécessaires pour les opérations ultérieures.

`get_collection()`

- **Args:** Aucun
- **Returns:** Collection MongoDB active.
- **Description:** Méthode d'accès pour obtenir la référence à la collection active, permettant d'effectuer des opérations MongoDB directement si nécessaire.

`set_collection(collection_name)`

- **Args:**
  - `collection_name` (str): Nom de la nouvelle collection à utiliser.
- **Returns:** None
- **Description:** Met à jour la collection active dans la même base de données. Utile pour basculer entre différentes collections sans recréer une nouvelle instance de DatabaseManager.

`shut_client()`

- **Args:** Aucun
- **Returns:** None
- **Description:** Ferme proprement la connexion au client MongoDB pour libérer les ressources. Important pour éviter les fuites de connexion, particulièrement dans les applications à longue durée d'exécution.

`insert_data(data_dir, reset=False, format=True)`

- **Args:**
  - `data_dir` (str): Chemin vers le répertoire contenant les fichiers Excel.
  - `reset` (bool): Si True, vide les collections avant d'insérer les nouvelles données.
  - `format` (bool): Si True, applique le formatage des données avant insertion.
- **Returns:** None
- **Description:** Parcourt tous les fichiers Excel dans le répertoire spécifié. Pour chaque fichier, la première feuille est considérée comme les données principales et est insérée dans la collection principale. Les autres feuilles sont considérées comme des tables r et sont insérées dans une collection dédiée. Le champ 'database' est automatiquement ajouté à partir du nom du dossier parent.

`__clean_value(v, key=None)`

- **Args:**
  - `v`: Valeur à nettoyer.
  - `key` (str, optionnel): Clé associée à la valeur, pour appliquer des règles spécifiques.
- **Returns:** Valeur nettoyée et standardisée.
- **Description:** Fonction privée utilisée pour le nettoyage des données. Applique diverses transformations selon le type de données et la clé : gestion des valeurs spéciales, normalisation des chaînes, corrections techniques, standardisation des codes et abréviations.

`__format_data(data)`

- **Args:**
  - `data` (list): Liste de dictionnaires représentant les enregistrements à formater.
- **Returns:** Liste des enregistrements formatés.
- **Description:** Fonction privée utilisée avant insertion dans la base de données. Nettoie chaque champ, ajoute des champs supplémentaires (environnement, usage, catégorisation d'âge, couleur des granulats, etc.).

`clean_mongo()`

- **Args:** Aucun
- **Returns:** None

- ∗ **Description:** Parcourt tous les documents de la collection active, applique les règles de nettoyage à chaque champ, puis met à jour les documents. Utile pour standardiser une base existante sans la recréer.

`get_data(query=None, key=None, r_table=False)`

- ∗ **Args:**
  - · `query` (dict, optionnel): Dictionnaire de filtre MongoDB.
  - · `key` (str ou list, optionnel): Champ ou liste de champs à inclure dans le résultat.
  - · `r_table` (bool): Si True, retourne aussi les tables r associées.
- ∗ **Returns:** Liste de documents ou tuple (documents, tables r).
- ∗ **Description:** Effectue une requête sur la collection principale et récupère les documents correspondants. Si `r_table=True`, recherche également les tables r associées à chaque document dans la collection `table_r` et les renvoie dans le même ordre. Si `key` est None, tous les champs sont récupérés.

`write_key(which='all')`

- ∗ **Args:**
  - · `which` (str): 'all' pour toutes les clés, 'usefull' pour une liste prédéfinie.
- ∗ **Returns:** None
- ∗ **Description:** Génère un fichier CSV contenant, pour chaque combinaison clé-valeur : le nom de la clé, la valeur unique, le nombre d'occurrences, le nombre de documents "réalistes" ayant cette valeur, et le pourcentage d'occurrences. Aide à analyser la distribution des valeurs dans la base.

`collection_stats()`

- ∗ **Args:** Aucun
- ∗ **Returns:** None
- ∗ **Description:** Calcule et affiche diverses statistiques sur la collection : nombre total d'entrées, taille de la base, répartition des types de données par champ, proportion globale, nombre de champs uniques.

`benchmark_queries(queries, keys, n_repeat=10)`

- ∗ **Args:**
  - · `queries` (list): Liste de dictionnaires représentant des requêtes MongoDB.
  - · `keys` (dict): Projection des champs à retourner.
  - · `n_repeat` (int): Nombre de répétitions pour calculer la moyenne.

- **Returns:** Liste de tuples (requête, temps moyen d'exécution).
- **Description:** Exécute chaque requête plusieurs fois et calcule le temps moyen d'exécution. Affiche et retourne les résultats sous forme de liste de tuples. Utile pour optimiser les performances et comparer différentes stratégies de requête.

`get_duplicate_entries(ignore_fields=None, r_table=True)`

- **Args:**
  - `ignore_fields` (list, optionnel): Liste de champs à ignorer lors de la comparaison.
  - `r_table` (bool): Si True, retourne aussi les tables r associées.
- **Returns:** Liste de groupes de documents similaires, ou tuple (groupes, tables r).
- **Description:** Recherche les documents identiques en ignorant certains champs spécifiés, puis les regroupe par similarité. Si r_table=True, retourne également les tables r associées aux groupes de doublons.

Standalone Utility Function

`export_data_and_rtables_to_excel(data, r_tables, file_name)`

- **Args:**
  - `data` (list): Liste de dictionnaires (documents principaux).
  - `r_tables` (list): Liste de matrices (ou None), dans le même ordre que data.
  - `file_name` (str): Nom du fichier Excel à créer.
- **Returns:** None
- **Description:** Fonction utilitaire pour exporter des données et leurs tables r associées vers un fichier Excel. Les données principales sont placées dans la feuille "main", et chaque table r dans une feuille séparée nommée d'après son identifiant.

User Guide for the `Data` Class

General Overview

extttData est une classe pour manipuler et visualiser des données extraites d'une base MongoDB, stockées sous forme de liste de dictionnaires. Elle fournit des méthodes pour regrouper, transformer et représenter graphiquement les données.

Class Methods

`__init__(data, collection)`

- **Args:**
  - `data` (list): Liste de dictionnaires contenant les données à analyser.
  - `collection`: L'objet collection MongoDB associé aux données.
- **Returns:** None
- **Description:** Initialise un objet Data en stockant les données et la référence à la collection MongoDB. Ces attributs sont utilisés par les autres méthodes pour manipuler et visualiser les données.

`get_data()`

- **Args:** Aucun
- **Returns:** Liste de dictionnaires contenant les données.
- **Description:** Méthode d'accès pour récupérer les données actuellement stockées dans l'instance. Utile après des transformations ou regroupements effectués par d'autres méthodes.

`set_data(data)`

- **Args:**
  - `data` (list): Nouvelle liste de dictionnaires à stocker.
- **Returns:** None
- **Description:** Remplace les données actuelles par un nouveau jeu de données. Généralement utilisée en interne par les méthodes qui transforment les données, comme split_by_age ou divide_by_key.

`split_by_age(limits)`

- **Args:**
  - `limits` (list): Liste d'entiers/floats croissants représentant les bornes supérieures des intervalles d'âge.
- **Returns:** None
- **Description:** Divise les données en groupes selon l'âge de la surface routière. Les limites fournies définissent les bornes des intervalles. Par exemple, avec limits = [0, 12, 24, 36, 48], les données seront réparties en 6 groupes : (¡0), [0,12), [12,24), [24,36), [36,48), et [48,+inf). Les documents sans âge valide sont ignorés.

`divide_by_key(key)`

- **Args:**
  - `key` (str): Clé des dictionnaires de données sur laquelle effectuer le regroupement.
- **Returns:** Liste de listes, chaque sous-liste contenant les documents (dicts) de self.data partageant la même valeur pour la clé spécifiée.
- **Description:** Regroupe l'ensemble de données en sous-listes selon les valeurs uniques de la clé spécifiée. Chaque sous-liste contient les éléments correspondant à une valeur particulière de la clé donnée.

`b_plot(labels=None, save=None, photo=None, bins=[0.47, 1.10, 3.43])`

- **Args:**
  - `labels` (list, optionnel): Liste d'étiquettes pour identifier chaque groupe de données. Si None, utilise "data1", "data2", etc.
  - `save` (str, optionnel): Nom du fichier pour sauvegarder le graphique au format PDF. Si None, n'enregistre pas le graphique.
  - `photo` (str, optionnel): Contrôle quels paramètres visualiser : None (S1 et Q0), "S1" ou "Q0".
  - `bins` (list, optionnel): Valeurs de référence pour tracer des lignes horizontales sur le graphique S1. Par défaut [0.47, 1.10, 3.43].
- **Returns:** None
- **Description:** Crée des boxplots pour visualiser la distribution statistique des paramètres S1 et/ou Q0 dans chaque groupe de données. Les boîtes à moustaches utilisent les percentiles 10 et 90 comme limites, la moyenne est marquée par une croix rouge. Pour S1, des lignes de référence sont tracées aux valeurs spécifiées dans bins. Le nombre d'éléments dans chaque groupe est indiqué dans les étiquettes.

`plot(categories=None, save=None, x_var="S1", y_var="Q0", bins=[0.47, 1.10, 3.43])`

- **Args:**
  - `categories` (list, optionnel): Liste de clés pour créer un graphique par catégorie. Si None, crée un seul graphique sans distinction de catégories.
  - `save` (str, optionnel): Nom du fichier pour sauvegarder le graphique au format PDF. Si None, n'enregistre pas le graphique.
  - `x_var` (str, optionnel): Nom de la variable à afficher sur l'axe des abscisses. Par défaut "S1".

    - `y_var` (str, optionnel): Nom de la variable à afficher sur l'axe des ordonnées. Par défaut "Q0".
    - `bins` (list, optionnel): Valeurs de référence pour tracer des lignes verticales ou horizontales selon la variable S1. Par défaut [0.47, 1.10, 3.43].
- **Returns:** None
- **Description:** Crée des graphiques de dispersion montrant la relation entre deux variables (par défaut S1 et Q0). Si des catégories sont spécifiées, les points sont colorés selon leurs valeurs pour ces catégories, et un graphique est créé pour chaque catégorie. Pour chaque groupe, le barycentre est indiqué par un cercle noir. Des lignes de référence sont tracées aux valeurs spécifiées dans bins (verticales si x_var="S1", horizontales si y_var="S1"). La légende indique le nombre de points par sous-catégorie.

User Guide for the `apprentissage.py` module

General Overview

Ce module regroupe les fonctions et classes pour l'apprentissage automatique, la gestion des modèles, l'évaluation et la visualisation des résultats sur les données routières.

Fonctions principales

`donnee_apprentissage(database, querie, key, photo='S1', one_hot=True, random_stat`
`test_size=0.2)`

- **Args:**
    - `database`: Instance de DatabaseManager.
    - `querie`: Dictionnaire de filtres pour la base de données.
    - `key`: Liste des colonnes/features à extraire.
    - `photo` (str, optionnel): 'S1' ou 'Q0' (par défaut 'S1').
    - `one_hot` (bool, optionnel): Si True, applique un encodage one-hot sur les colonnes de type string.
    - `random_state` (int, optionnel): Graine aléatoire pour la reproductibilité.
    - `test_size` (float, optionnel): Proportion des données pour le test (par défaut 0.2).
- **Returns:** X_train, X_test, y_train, y_test (ou X_train, y_train si test_size=0 ou 1)

- **Description:** Récupère et prépare les données d'apprentissage à partir de la base MongoDB, applique le prétraitement et l'encodage, puis effectue un split train/test selon les paramètres.

`accuracy_score_binned(y_true, y_pred, bins)`

- **Args:**
  - `y_true`: Valeurs réelles.
  - `y_pred`: Valeurs prédites.
  - `bins`: Intervalles pour le binning.
- **Returns:** Score d'accuracy binned (float).
- **Description:** Calcule l'accuracy entre y_true et y_pred en les binant selon les intervalles fournis.

Classe `Model`

`__init__(model, database, features, querie=, params=None)`

- **Args:**
  - `model`: Classe du modèle sklearn (RandomForestRegressor, HistGradientBoostingRegressor, etc.).
  - `database`: Instance de DatabaseManager.
  - `features`: Liste des features utilisées.
  - `querie` (dict, optionnel): Filtres pour la base de données.
  - `params` (dict, optionnel): Paramètres du modèle.
- **Returns:** None
- **Description:** Initialise un objet Model avec le modèle sklearn, la base de données, les features et les paramètres.

`set_model(model, params=None)`

- **Args:**
  - `model`: Classe du modèle sklearn.
  - `params` (dict, optionnel): Paramètres du modèle.
- **Returns:** None
- **Description:** Définit le modèle sklearn utilisé par l'objet Model.

`set_features(features)`

- **Args:**
  - `features`: Liste de chaînes de caractères.
- **Returns:** None
- **Description:** Définit la liste des features utilisées par le modèle.

`set_querie(querie)`

- **Args:**
  - `querie`: Dictionnaire de filtres.
- **Returns:** None
- **Description:** Définit le dictionnaire de filtres pour la base de données.

`get_querie()`

- **Args:** Aucun
- **Returns:** Dictionnaire de filtres.
- **Description:** Retourne le dictionnaire de filtres utilisé.

`get_model()`

- **Args:** Aucun
- **Returns:** Modèle sklearn utilisé.
- **Description:** Retourne le modèle sklearn utilisé.

`get_features()`

- **Args:** Aucun
- **Returns:** Liste des features utilisées.
- **Description:** Retourne la liste des features utilisées.

`get_params()`

- **Args:** Aucun
- **Returns:** Dictionnaire des paramètres du modèle.
- **Description:** Retourne les paramètres du modèle sklearn utilisé.

`fit_model(X_train, y_train)`

- **Args:**
  - `X_train` (pd.DataFrame): Données d'entraînement.
  - `y_train` (pd.Series): Cible d'entraînement.
- **Returns:** None
- **Description:** Entraîne le modèle avec les données d'entraînement.

`predict(X)`

- **Args:**
  - `X` (pd.DataFrame): Données pour la prédiction.
- **Returns:** np.ndarray: Prédictions du modèle.
- **Description:** Fait des prédictions avec le modèle entraîné.

`unified_grid_search(...)` **(voir code)**

- **Args:** Nombreux (voir code).
- **Returns:** Dictionnaire avec les meilleures features, meilleurs paramètres et scores.
- **Description:** Recherche optimale des paramètres ou des features (validation croisée, OOB, etc.).

`evaluate_binned(bins, y_true=None, y_pred=None, labels=None, output_dict=False, data=None)`

- **Args:**
  - `bins` (array-like): Liste des seuils pour la classification.
  - `y_true` (array-like, optionnel): Valeurs réelles à classifier.
  - `y_pred` (array-like, optionnel): Valeurs prédites à classifier.
  - `labels` (array-like, optionnel): Étiquettes pour chaque classe.
  - `output_dict` (bool): Si True, retourne un dictionnaire avec les métriques.
  - `data` (dict, optionnel): Dictionnaire contenant X_test et y_test (non utilisé ici).
- **Returns:** Dictionnaire ou tuple (rapport, exactitude).
- **Description:** Évalue les performances du modèle en transformant les prédictions continues en classes selon des intervalles ("bins"). Calcule les métriques de classification pour chaque intervalle et l'exactitude globale.

`plot_predictions(y_true, y_pred, ...)` **(voir code)**

- **Args:** Nombreux (voir code).
- **Returns:** matplotlib.figure.Figure
- **Description:** Affiche un scatter plot comparant les valeurs réelles et prédites avec des lignes de délimitation et zones de bonnes prédictions.

`shap_explainer()`

- **Args:** Aucun
- **Returns:** Un objet TreeExplainer de SHAP.
- **Description:** Retourne un explainer SHAP pour le modèle courant.

`cross_val_evaluation(X, y, bins, cv=5, random_state=42)`

- **Args:**
  - `X`: Features (DataFrame ou ndarray).
  - `y`: Cibles (Series ou ndarray).
  - `bins`: Intervalles pour le calcul de l'accuracy.
  - `cv` (int): Nombre de folds.
  - `random_state` (int): Pour la reproductibilité.
- **Returns:** Dictionnaire avec les scores par fold et les moyennes pour accuracy, r2 et rmse.
- **Description:** Calcule l'accuracy moyenne binned, le R^2 et le RMSE sur différents plis de validation croisée.