MASTER'S DEGREE THESIS

Master of Science in Advanced Mathematics and Mathematical Engineering

Title: Development of predictive flight fuel consumption models

Author: García Hernández, Sílvia

Advisor: de Villardi de Montlaur, Adeline

Department: Department of Physics

Academic year: 2023-2024

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Master in Advanced Mathematics and Mathematical Engineering

Master's thesis

Development of predictive flight fuel consumption models

Sílvia García Hernández

Supervised by Adeline de Villardi de Montlaur

Facultat de Matemàtiques i Estadística
Universitat Politècnica de Catalunya
Q1 2023-2024

I would like to thank my advisor Adeline for this thesis proposal and all her supervision and guidance during the project. I also thank professor Josep Ginebra for his help.

Finally, I would like to show my gratitude to my family and friends. A special thank you to Mercè for helping and cheering me up, and to Edgar, for encouraging and supporting me throughout my studies.

Abstract. Fuel consumption is a crucial consideration in the aviation industry. This last one, integral to global connectivity, faces significant environmental concerns due to the escalating demand for air travel and its substantial contribution to greenhouse gas emissions.

In this thesis, we present predictive models calculating gate-to-gate fuel consumption, using simple variables such as flight distance, and taking into account the available number seats for each aircraft, in contrast with other flight consumption calculators. The main goal of this work is to construct an indicator that can be used to compare emissions with other travel alternatives. Specifically, we develop the theoretical framework for the presented models and showcase their results. Using the model with best accuracy, a LightGBM model, we demonstrate a real-world application by conducting a CO_2 emission comparison between flight and rail routes.

Contents

Nomenclature

ICAO	International Civil Aviation Organization
MIDT	Market Information Data Tapes
GCD	Great Circle distance
IMPACT	Integrated Aircraft Noise and Emissions Modelling Platform
MTOM	Maximum Take Off Mass
SVD	Singular Value Decomposition
ASK	Available Seats per Kilometer
MSE	Mean Squared Error
LightGBM	Light Gradient-Boosting Machine
MAPE	Mean Absolute Percentage Error
PAX	Passenger
IATA	International Air Transport Association

Introduction

In an era dominated by technological advances and rapid globalization, the aviation industry plays a pivotal role in connecting the world. However, the constant demand for air travel has raised significant environmental concerns, contributing substantially to greenhouse gas emissions. While other industries, including within the transport sector, might be able to substantially decarbonise, aviation faces serious challenges with a current focus on technological solutions [1, 2], sustainable aviation fuels [3] and carbon compensation mechanisms [4, 5].

This concern is evident in various contemporary situations. For instance, in the recent investiture of Pedro Sánchez, the current prime minister of Spain, the agreement with another political party played a key role. This agreement included the provision to eliminate short flights if a train alternative of no more than 2.5 hours was available [6], thereby contributing to the reduction of greenhouse gas emissions. Furthermore, major airline companies are now offering integrated tickets to circumvent short-haul flights, replacing them with railway transport when stopovers are required [7].

To comprehensively evaluate the impact of different means of transport and alternative travel options, the consideration of indicators taking into account both emissions and the number of passengers is essential. However, currently available models to assess aviation emissions such as the ICAO *Carbon Emissions Calculator* [8], tend to focus on simple linear relationships between distance and/or time and emissions or to be based on detailed fuel consumption models, which require detailed parameters (e.g., fully defined trajectories) on a flight-by-flight basis [9].

In this thesis, we present predictive models calculating gate-to-gate fuel consumption, using simple variables such as flight distance, and taking into account the available number seats for each aircraft, in contrast with other flight consumption calculators. The main goal of this work is to construct an indicator that can be used to compare emissions with other travel alternatives, such as the rail one. Specifically, we develop the theoretical framework for the presented models, showcase their results, and demonstrate a real-world application by conducting a CO_2 emission comparison between flight and rail routes.

Chapter 1

Data and objectives

This section will dive into the data and modelling objectives. We will start by outlining our model objectives, followed by a detailed description of our data, including its source and content. Subsequently, we will allocate multiple sections to comprehensively address the data-preprocessing.

1.1 Modelling objectives

The main goal of this thesis is to build a model that can accurately calculate gate-to-gate flight fuel consumption. Previous research indicates that extensive information is not necessarily required to derive fuel consumption estimates [9]. In contrast to other calculators, such as the International Civil Aviation Organization (ICAO) *Carbon Emissions Calculator* [8], the model will take into account the available seats of each aircraft. This addition is crucial for making fair comparisons of fuel consumption per passenger across other travel alternatives. Additionally, it is worth noting that once fuel consumption is known, calculating emissions such as CO_2 becomes a straightforward process, making this type of model useful to compute flight emissions.

1.2 Data and data preprocessing

The dataset used in this analysis originates from the Market Information Data Tapes (MIDT) for Traffic Data, specifically capturing historical records of operations conducted to or from Europe during the calendar year 2017. Table 1.1 describes the dataset columns of the presented dataset.

In particular, it contains 58125 different routes (number of rows) and 91 different aircraft models. Since it contains all routes to or from Europe, we have a very wide interval for the distance traveled, being 100 kilometers the minimum distance and 11961 kilometers the maximum one.

Column Name	Description
Origin_Airport	Origin Airport
Destination_Airport	Destination Airport
_Fleet	Aircraft type, using the ICAO designation
_Seats_per_Operation	Number of available seats
Distance_(km)	Traveled distance, using the corresponding Great Circle Distance (GCD) approximation
_Depcount	Number of times that the flight took place during the year 2017
FUEL_BURNT_KG	Kilograms of fuel burn, obtained from IMPACT, the web-based modelling platform from EUROCONTROL

Table 1.1: Column description of the dataset

As we can see on Figure 1.1, our data is not equally distributed. As one can assume, short flights are more present compared to long-distance ones: the median, shown with a white dot is located at 1286 km. This distance distribution of the data will add complexity to our analysis.

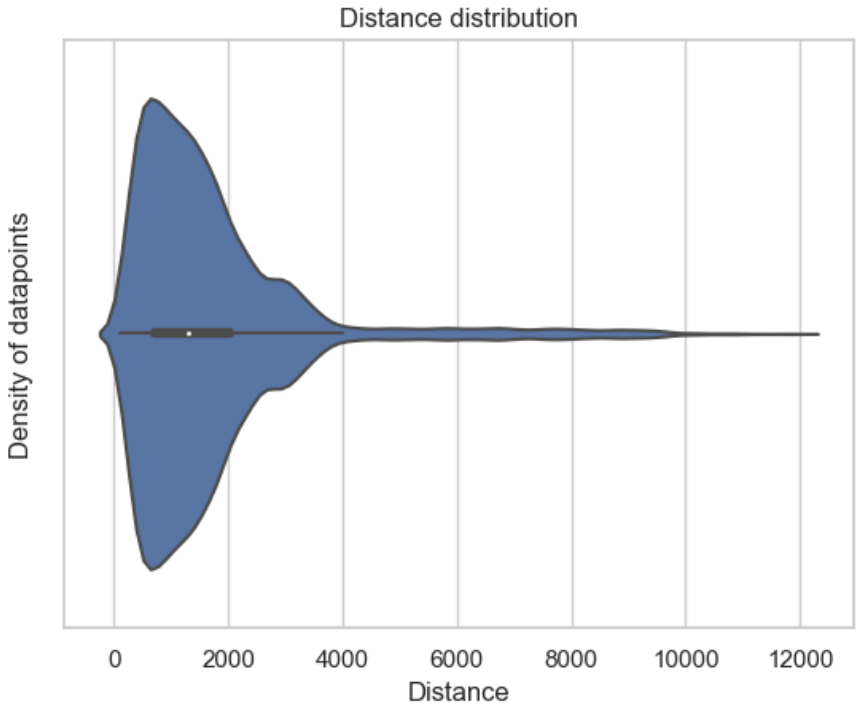


Figure 1.1: Violin plot of the distribution of flight distances in our dataset

1.2.1 GCD and corrections

The Integrated aircraft noise and emissions modelling platform (IMPACT) software estimates the flight consumption from take-off to landing considering the Great Circle Distance (GCD) between origin and arrival airports, which corresponds to the shortest distance between two points on the surface of a sphere, measured along the surface of the sphere. Note that a correction factor must be added due to different air traffic procedures, such as air traffic congestion or routing deviations. ICAO suggests a factor correction based on the distance [10] shown in Table 1.2.

GCD	Correction to GCD
Less than 550 Km	+ 50 km
Between 550 km and 5500 km	+ 100km
Above 5500 km	+ 125 km

Table 1.2: GCD factor correction from ICAO methodology [10]

As we can appreciate in Table 1.2, this correction does not follow a proportional increase. For instance, flights with distances falling between 550 and 5500 kilometers, a very wide range in which more than 78% of our data falls, experience a correction of 100 kilometers, no matter their flight distance. In order to make this correction continuous, we decided to consider the *Performance Review Report of 2022* [11], where it is stated that the horizontal en-route flight inefficiency is of 3.08%[1]. This coefficient is computed taking the additional distance flight with respect to the GCD. Thus, we applied

$$\text{Corrected distance (km)} = \frac{1}{1.0308} \cdot \text{Distance (km)} \tag{1.1}$$

in order to obtain a better estimation of the flight distance.

1.2.2 Fuel and taxi time

In aviation, taxiing is defined as the movement of an aircraft on the ground, under its own power [12]. Taxi time and its corresponding fuel consumption accounts for emissions on-ground. Even if for larger flights this coefficient could be considered negligible due to high fuel consumption, it might contribute significantly to the total fuel consumption for short flights. Thus, these emissions should be considered.

We define the taxi-in and taxi-out times as the duration between landing time and gate in time, and between gate out time and take off time, respectively, illustrated in 1.2. These intervals of time may differ in each route due to gate location or airport layout and configuration, between others. In order to be able to measure this consumption, we need the taxi

[1]Vertical inefficiency was considered negligible for this correction.

times and fuel consumption during this process.

We used the taxi times provided by EUROCONTROL [13], from which the mean taxi-in/out time in minutes was available for almost every airport in our dataset. For airports without information, the mean taxi/out times of all airports was considered, which was 5.76 and 12.15 minutes, respectively.

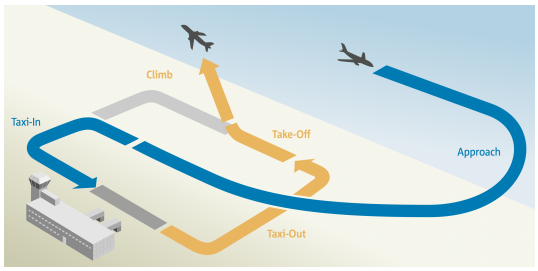


Figure 1.2: Taxi-in and taxi-out illustration. Source: EUROCONTROL

Fuel consumption during taxiing is a factor that varies depending on the aircraft model and its number of engines. It is obvious that larger airplanes with a greater number of engines result in higher fuel consumption. To provide accurate data, we have tried to find specific values for each aircraft model in our database rather than relying on an average estimate. Nevertheless, this information is not always available. For some of our models, we have been able to find the taxi fuel consumption per second (kg/s), which we will call `Available Fuel`. For others, estimations were necessary.

For models where precise taxi fuel consumption (`Available Fuel`) data is lacking, a classification into 6 different categories was performed, following the RECAT-EU classification [14], based on the certificated Maximum Take Off Mass (MTOM) and the span of the aircraft. Within each category, we compute the mean `Taxi Fuel` consumption, drawing from available information (`Available Fuel`) on models falling into the respective category. `Total` fuel consumption during taxi is then determined using the formula

$$\texttt{Total} = 60 \cdot (\texttt{Taxi-in} + \texttt{Taxi-out}) \cdot \texttt{Taxi Fuel}. \tag{1.2}$$

Model	CAT	Origin	Dest	Taxi in	Taxi out	Available Fuel	Taxi Fuel	Total
A320	D	EIDW	EGLL	8.05	14.45	0.212	0.212	282.38
A345	B	LEMD	MPTO	6.46	18.58	0.635	0.635	954.02
AT72	E	LESO	LEMD	8.97	8.21	-	0.179	184.51
B737	D	UGTB	UUWW	9.73	10.69	-	0.219	268.318

Table 1.3: Table illustrating the procedure of taxi correction. `Taxi-in` and `Taxi-out` in minutes, `Available Fuel`, `Taxi Fuel`, and `Total` expressed in kg/second

As depicted in Table 1.3, when `Available Fuel` is present, it serves as the `Taxi Fuel` coefficient in (1.2)—the taxi fuel consumption per second. Conversely, when this information is not available, the mean of aircraft models with informed `Available Fuel` for each category `CAT` is employed. Taxi fuel consumption for each aircraft model of our database can be found in Appendix A.

Chapter 2

Methodology

Machine learning is a subfield of artificial intelligence that focuses on the development of algorithms and statistical models that enable computer systems to improve their performance on a specific task over time, without being explicitly programmed. Its essence lies in the ability to learn from data and make decisions without human intervention. Among its diverse set of techniques, we find *Supervised Learning*, one of its fundamental paradigms.

Supervised Learning is a type of machine learning where the algorithm is trained on a labeled dataset, comprising input-output pairs. This approach equips the algorithm to learn a mapping function from input features to corresponding output labels. This paradigm is particularly adept at tasks like classification and regression, being thus suitable for our problem, where we want to map some features, like the distance or the number of seats to the fuel consumption. Note that some of the classical methods used, such as linear regression, are indeed simple supervised machine learning models.

In this section we will present the methods used to try to model our dataset. In order to set a notation, we will consider a data distribution $(X, y) \sim \mathbb{P}_{(X,y)}$, where $X \in \mathbb{R}^d$ is the feature vector and $y \in \mathbb{R}$ the dependent variable we want to predict, that we will also call label. We will assume that we are given n independent and identically distributed training pairs $\{X^{(i)}, y^{(i)}\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$. Our goal is to find functions $\tilde{f} \in \mathcal{F} = \{f : \mathbb{R}^d \longrightarrow \mathbb{R}\}$ that are capable of obtaining good predictions in unseen samples.

In the first and second sections of this chapter, we will be will be following the notes from the course *Modern Machine Learning: Simple Methods that Work* [15], by Adityanarayanan Radhakrishnan, at MIT.

2.1 Linear Regression

Let us begin by presenting one of the most basic ways to model our data: *Linear Regression*. This method tries to find the line that best fits our training data, by splitting the response variable y_i, into a part that can be explained through linear combination of the explanatory variables X_i, also called signal, and a part that can not be explained, ε_i, known as noise. Thus, in linear regression we would like to find coefficients $\theta \in \mathbb{R}^{d+1}$ such that

$$y^{(i)} = \theta_0 + \theta_i X_1^{(i)} + \cdots + \theta_d X_d^{(i)} + \varepsilon^{(i)}, \qquad \text{for } i \in \{1, 2, \ldots, d\} \tag{2.1}$$

In particular, once we have found a model, in order to be able to apply some theoretical results we will need to check if our model verifies the hypothesis of a *normal lineal model*:

(1) It is linear, $E\left(y \mid \{X_j\}_{j\in[d]}\right) = \theta_0 + \theta_1 X_1 + \ldots + \theta_d X_d$

(2) The variance is constant, $V\left(\varepsilon^{(i)} \mid y^{(i)}\right) = \sigma^2$, (same value of σ^2 for all i)

(3) All $\varepsilon^{(i)}$ are normally distributed

(4) For all $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$ the residuals $\varepsilon^{(i)}, \varepsilon^{(j)}$ are independent

These assumptions are based on some observations. Firstly, by the Central Limit Theorem, any randomly distributed error should converge to a normal distribution when enough samples are taken.. On the other hand, the constant variance assumption aims to prevent scenarios where the approximation is effective only within a specific range of predicted values but fails to accurately represent other segments. Consequently, there could be a situation where a linear approximation works well in one part of the data but lacks accuracy in another segment.

Returning to Equation (2.1), our main objective will be to find the best value θ, that we will call $\tilde{\theta}$, such that the error ε is minimized. To do so, we need to fix a certain loss function and an algorithm that will minimize this loss function to find the optimal values for the θ coefficients. In particular, from Equation (2.1), our function $\tilde{f}$ will have the form $\tilde{f}(X) = \tilde{\theta}X$.

In order to develop the following ideas, we will fix a classical loss function, the *mean squared error*:

Definition 2.1 (MSE)**.** Let y be the actual value and $\tilde{y}$ be the predicted one. The mean squared error (MSE) between y and $\tilde{y}$ is $\mathcal{L}(y, \tilde{y}) = \frac{1}{2}\|y - \tilde{y}\|_2^2$.

Now, rewriting it using the notation of Equation (2.1), we will find $\tilde{\theta}$ by minimizing

$$\mathcal{L}(\theta) = \frac{1}{2}\sum_{i=1}^{n}(y^{(i)} - \theta X^{(i)})^2. \tag{2.2}$$

The only thing that now remains is to set an algorithm that is able to minimize (2.2). Even if there are other algorithms that are able to give a closed-form solution, we will choose *gradient descent* to continue presenting the following concepts.

Definition 2.2 (Gradient Descent)**.** Given a loss function $\mathbb{L}(\theta) : \mathbb{R}^d \longrightarrow \mathbb{R}$, an initial value $\theta_0 \in \mathbb{R}$ and $\lambda \in \mathbb{R}$, the step size, also called learning rate, we define the gradient descent algorithm as

$$\theta^{(t+1)} = \theta^{(t)} - \lambda \nabla_\theta \mathbb{L}(\theta^{(t)}) \qquad t \in \mathbb{Z}_+, \tag{2.3}$$

that minimizes the loss function $\mathbb{L}(\theta)$.

Let us now prove that for $t \to \infty$, we can identify the largest learning rate λ such that (2.3) converges to a certain limit point. Before that we provide a definition of Singular Value Decomposition (SVD) that will be used.

Definition 2.3 (SVD)**.** Given a $m \times n$ real matrix M, a Singular Value Decomposition is the factorization of the matrix M into $M = U\Sigma V^t$, where U, V are orthogonal matrices with sizes $r \times r$ and $n \times n$, and Σ is a $r \times n$ diagonal matrix with non-negative entries in the diagonal.

It is a folklore theorem that such a decomposition always exists. With this definition we are now able to prove the following theorem:

Theorem 2.4. *Let $X = (X^{(1)}, X^{(2)}, ..., X^{(n)})$ and $y = (y^{(1)}, y^{(2)}, \dots, y^{(n)})$. Let σ_1 be the largest singular value of X. Initializing $\theta^{(0)} = 0$ and using gradient descent, the minimization of the loss function*

$$\mathcal{L}(\theta) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \theta X_i)^2,$$

converges to a global minimum, $\hat{\theta} = yX^\dagger$, for $0 < \lambda < \frac{2}{\sigma_1^2}$, where $X^\dagger$ is the Moore-Penrose pseudoinverse of X.

Proof. Starting from gradient descent, we can write

$$\theta^{(t+1)} = \theta^{(t)} - \lambda \nabla_\theta \mathbb{L}(\theta^{(t)}) = \theta^{(t)} - \lambda(y - \theta X)X^T.$$

Let $M = XX^T$ and $M' = yX^T$. Thus, we have

$$\theta^{(t+1)} = \theta^{(t)}(I - \lambda M) + \lambda M'.$$

By induction, let us prove that $\theta^{(t)} = \lambda M'[(I-\lambda M)^{t-1} + (I-\lambda M)^{t-2} + \cdots + (I-\lambda M)^1 + I]$. For $\theta^{(0)} = 0$, it is trivially true. Let us suppose that it is true for $\theta^{(t)}$ and prove it for $\theta^{(t+1)}$:

$$\begin{aligned}
\theta^{(t+1)} &= \theta^{(t)}(I - \lambda M) + \lambda M' \\
&= (\lambda M'[(I - \lambda M)^{t-1} + (I - \lambda M)^{t-2} + \cdots + (I - \lambda M)^1 + I])(I - \lambda M) + \lambda M' \\
&= \lambda M'[(I - \lambda M)^t + (I - \lambda M)^t + \cdots + (I - \lambda M)^1 + I].
\end{aligned}$$

With this recurrence, consider the singular value decomposition of our matrix $X = U\Sigma V^T$, with $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$. Rewriting, we obtain:

$$M = (U\Sigma V^T)(U\Sigma V^T)^T = U\Sigma^2 U^T, \qquad M' = yV\Sigma U^T.$$

Thus, the recurrence can be rewritten as follows:

$$\theta^{(t)} = \lambda M' U \Sigma^+ U^T,$$

where $\Sigma^+ = (I - \lambda\Sigma^2)^{t-1} + (I - \lambda\Sigma^2)^{t-2} + \cdots + (I - \lambda\Sigma^2)^1 + I$ is a diagonal matrix with the r first diagonal entries forming a geometric sum for $\lambda < 2/\sigma_1^2$:

$$\Sigma_i^+ = \sum_{k=1}^{t-1}(1 - \lambda\sigma_1^2)^k = \frac{1 - (1 - \lambda\sigma_i^2)t}{\lambda\sigma_1^2},$$

resulting in

$$\Sigma^+ = \begin{bmatrix} \frac{1-\left(1-\lambda\sigma_1^2\right)^t}{\lambda\sigma_1^2} & 0 & \ldots 0 & \\ 0 & \frac{1-\left(1-\lambda\sigma_2^2\right)^t}{\lambda\sigma_2^2} & \ldots 0 & \mathbf{0}_{r\times d-r} \\ 0 & \ldots & \frac{1-\left(1-\lambda\sigma_r^2\right)^t}{\lambda\sigma_r^2} & \\ & \mathbf{0}_{d-r\times r} & & t\mathbf{I}_{d-r\times d-r} \end{bmatrix} \tag{2.4}$$

To finish, substituting M' by $yV\Sigma U^T$, we obtain

$$\theta^{(t)} = yV\Sigma^\dagger U^T \text{ with } \Sigma^\dagger = \begin{bmatrix} \frac{1-\left(1-\lambda\sigma_1^2\right)^t}{\sigma_1} & 0 & \ldots & \\ 0 & \frac{1-\left(1-\lambda\sigma_2^2\right)^t}{\sigma_2} & \ldots & \mathbf{0}_{r\times d-r} \\ 0 & \ldots & \frac{1-\left(1-\lambda\sigma_r^2\right)^t}{\sigma_r} & \\ & \mathbf{0}_{d-r\times r} & & \mathbf{0}_{d-r\times d-r} \end{bmatrix}$$

Finally, taking $t \to \infty$, we can write

$$\theta^{(\infty)} = \lim_{t\to\infty} \theta^{(t)} = yV\Sigma^\dagger U^T \text{ where } \Sigma^\dagger = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & \ldots & \\ 0 & \frac{1}{\sigma_2} & \ldots & \mathbf{0}_{r\times d-r} \\ 0 & \ldots & \frac{1}{\sigma_r} & \\ & \mathbf{0}_{d-r\times r} & & \mathbf{0}_{d-r\times d-r} \end{bmatrix}$$

as we wanted. □

An important result that Theorem 2.4 gives us for $\theta^{(0)} = 0$ is that, for all $t \in \mathbb{N}$, $\theta^{(t)}$ can be written as the linear combination of the columns of our training matrix X, and thus, the output of the training predictor always lies in the span of the training data. This note will be a basic concept to use in the following section. Let us prove this statement:

Proposition 2.5. *Let* $\left\{X^{(i)}\right\}_{i=1}^{n} \subset \mathbb{R}^d$ *and* $\left\{y^{(i)}\right\}_{i=1}^{n} \subset \mathbb{R}$. *Then there exist* $\{\alpha_i\}_{i=1}^{n} \subset \mathbb{R}$, *such that the minimum* ℓ_2 *norm minimizer,* $\tilde{\theta}$, *for the loss:*

$$\mathcal{L}(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(y^{(i)} - \theta X^{(i)}\right)^2$$

has the form

$$\hat{\theta} = \sum_{i=1}^{n} \alpha_i X^{(i)^T}. \tag{2.5}$$

Proof. Let us begin proving that $\tilde{\theta} = yX^{\dagger}$ is the minimum ℓ_2 norm solution. Let us define $d = \text{ rank}(X)$.

For **n=d**, we can declare that there is exactly one solution. In particular, solving the linear system given by $\theta = yX^{-1}$ gives us the exact solution. For this case, $X^{\dagger} = X^{-1}$.

For $\mathbf{n > d}$, we know that computing the inverse of X is not possible. Instead, we will find a solution that minimizes the MSE by setting the gradient of the MSE equal to zero:

$$\nabla_w \mathcal{L}(\theta) = 0 \Longrightarrow (y - wX)X^T = 0 \Longrightarrow w = yX^T\left(XX^T\right)^{-1}$$

Notice that XX^T is invertible since $n > d$ and the rank of X is d. Furthermore, substituting $X = U\Sigma V^T$ given by the singular value decomposition, we find that $yX^T\left(XX^T\right)^{-1} = yX^{\dagger}$.

For the last remaining case, $\mathbf{n < d}$, we know that there are infinitely many interpolating solutions to linear regression from linear algebra. Taking any of them will suffice the conditions of our statement, similarly to the $n = d$ case.

Let us continue by showing that the solution given by gradient descent has the form of Equation (2.5). To prove it, we will proceed using induction. Notice that for $\theta^{(0)}$ it is enough to set $\alpha_i = 0$ for all i. We will assume that Equation (2.5) holds for t and prove it for $t + 1$:

$$\theta^{(t+1)} = \theta^{(t)} - \lambda \nabla_\theta \mathbb{L}(\theta^{(t)}) = \theta^{(t)} + \lambda\left(y - \theta^{(t)}X\right)X^T$$

Now, rewriting $\left(y - \theta^{(t)}X\right)X^T = \sum_{i=1}^{n} \beta_i^{(t)} X^{(i)^T}$, it follows

$$\begin{aligned}
\theta^{(t+1)} &= \theta^{(t)} + \lambda \sum_{i=1}^{n} \beta_i^{(t)} X^{(i)^T} \\
&= \sum_{i=1}^{n}\left(\alpha_i^{(t)} + \lambda\beta_i^{(t)}\right)X^{(i)^T} \\
&= \sum_{i=1}^{n} \alpha_i^{(t+1)} X^{(i)^T},
\end{aligned}$$

as we wanted to show. □

2.2 Kernel Regression

In this section we will like to extend the notion of linear regression into a nonlinear one. Recall that an effective linear mapping from samples to labels rarely exists. Thus, we apply a nonlinear transformation to our training data X_i. To formally define this set of transformations, let us first present the space in which they are defined:

Definition 2.6 (Hilbert Space)**.** A *Hilbert Space* $\mathcal{H}$ is a real or complex inner product space that is also a complete metric space with respect to the distance function induced by the inner product.

Now, the functions we will apply to our training data belong to

$$\mathcal{F} := \{f : \mathbb{R} \longrightarrow \mathbb{R} \text{ such that } f(x) = \langle \theta, \psi(x) \rangle_{\mathcal{H}},\ \psi : \mathbb{R} \longrightarrow \mathcal{H},\ \theta \in \mathcal{H}\}$$

where $\mathcal{H}$ is a Hilbert space with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and ψ is a nonlinear feature map.

Usually, when selecting an appropriate feature transformation we are able to build an effective predictor, even while using linear functions. But how can we select an appropriate feature transformation for any given dataset? This may be a hard question to answer when there is no knowledge about useful features, and it is often beneficial to consider random feature maps into a Hilbert space.

As we know, from Preposition 2.5 $\tilde{\theta}$ can be written as in (2.5). We can repeat the same analysis by substituting X by the sample matrix $\psi(X)$, extending the prove to the case when the $\mathcal{H}$ is a general Hilbert space as follows:

Theorem 2.7 (Representer Theorem)**.** *Let $\mathcal{H}$ be a Hilbert space with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Let $\left\{\psi\left(X^{(i)}\right)\right\}_{i=1}^{n} \subset \mathcal{H}$ and $\left\{y^{(i)}\right\}_{i=1}^{n} \subset \mathbb{R}$. Then, there exist $\{\alpha_i\}_{i=1}^{n} \subset \mathbb{R}$, such that the minimum $\mathcal{H}$-norm minimizer, $\tilde{\theta}$, for the loss:*

$$\mathcal{L}(\theta) = \frac{1}{2} \sum_{i=1}^{n} \left(y^{(i)} - \left\langle \theta, \psi\left(X^{(i)}\right) \right\rangle_{\mathcal{H}}\right)^2 \tag{2.6}$$

has the form

$$\tilde{\theta} = \sum_{i=1}^{n} \alpha_i \psi\left(X^{(i)}\right). \tag{2.7}$$

Proof. We prove the result by first showing that all minimizers of the loss differ only by a term that is orthogonal to the transformed training samples, and then, we will use the Pythagorean theorem to show that the solution in the span of the transformed training samples has minimum norm.

Let us consider the orthogonal decomposition of a $\tilde{\theta} \in \mathcal{H}$ onto the space spanned by $\psi\left(X^{(i)}\right)$ and its complement. In particular, there exists some orthonormal basis $\{\phi_i\}_{i=1}^{n} \subset \mathcal{H}$ for

$\left\{\psi\left(X^{(i)}\right)\right\}_{i=1}^{n}$ and some $v \in \mathcal{H}$ orthogonal to all ϕ_i such that, by Proposition 2.5, we can write:

$$\tilde{\theta} = \sum_{i=1}^{n} \beta_i \phi_i + v$$

We thus have that:

$$\begin{aligned}
\mathcal{L}(\tilde{\theta}) &= \frac{1}{2}\sum_{i=1}^{n}\left(y^{(i)} - \left\langle \tilde{\theta}, \psi\left(X^{(i)}\right)\right\rangle_{\mathcal{H}}\right)^2 \\
&= \frac{1}{2}\sum_{i=1}^{n}\left(y^{(i)} - \left\langle \sum_{i=1}^{n}\beta_i\phi_i + v, \psi\left(X^{(i)}\right)\right\rangle_{\mathcal{H}}\right)^2 \\
&= \frac{1}{2}\sum_{i=1}^{n}\left(y^{(i)} - \left\langle \sum_{i=1}^{n}\beta_i\phi_i, \psi\left(X^{(i)}\right)\right\rangle_{\mathcal{H}}\right)^2 \\
&= \mathcal{L}(\tilde{\theta})
\end{aligned}$$

Hence, the loss does not change when adding a term orthogonal to the span of the $\psi\left(X^{(i)}\right)$. Now, by the Pythagorean theorem:

$$\|\tilde{\theta}\|_{\mathcal{H}}^2 = \left\|\sum_{i=1}^{n}\beta_i\phi_i + v\right\|_{\mathcal{H}}^2 = \left\|\sum_{i=1}^{n}\beta_i\phi_i\right\|_{\mathcal{H}}^2 + \|v\|_{\mathcal{H}}^2 \geq \left\|\tilde{\theta}\right\|_{\mathcal{H}}^2,$$

as we wanted to show. □

Once proved this Theorem, we can assert that for any $\psi : \mathbb{R}^d \to \mathcal{H}$, we can solve a linear regression problem in a Hilbert space, first finding the coefficients α_i and and finally using the representation for the minimum norm solution given by Theorem 2.7. This procedure is what we call *Kernel Regression*.

Recall that, for solving our kernel regression, we would like to solve the problem of minimizing the loss in Equation (2.6), which seems quite challenging. Using the result of Theorem 2.7, we will convert this problem into solving a finite dimensional linear regression one: instead of minimizing $\mathcal{L}(\theta)$ over all possible values of θ, we minimize the loss $\mathcal{L}$ with respect to the parameters $\{\alpha_i\}_{i=1}^{n}$. Thus, we have

$$\mathcal{L}(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(y^{(i)} - \left\langle \theta, \psi\left(X^{(i)}\right)\right\rangle_{\mathcal{H}}\right)^2.$$

Taking $\theta = \sum_{i=1}^{n}\alpha_i\psi\left(X^{(i)}\right)$, we can write

$$\mathcal{L}(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(y^{(i)} - \left\langle \sum_{j=1}^{n}\alpha_j \psi\left(X^{(j)}\right), \psi\left(X^{(i)}\right)\right\rangle_{\mathcal{H}}\right)^2$$

$$= \frac{1}{2}\sum_{i=1}^{n}\left(y^{(i)} - \begin{pmatrix}\alpha_1 & \alpha_2 & \dots & \alpha_n\end{pmatrix} \underbrace{\begin{pmatrix}\left\langle \psi\left(X^{(1)}\right), \psi\left(X^{(i)}\right)\right\rangle_{\mathcal{H}} \\ \left\langle \psi\left(X^{(2)}\right), \psi\left(X^{(i)}\right)\right\rangle_{\mathcal{H}} \\ \vdots \\ \left\langle \psi\left(X^{(n)}\right), \psi\left(X^{(i)}\right)\right\rangle_{\mathcal{H}}\end{pmatrix}}_{\hat{K}}\right)^2. \tag{2.8}$$

What Theorem 2.7 asserts is that finding the value of α_i for all i will yield the minimum $\mathcal{H}$-norm solution that minimizes the loss of Equation (2.7). What is important about this last equation is that it show us that we only need to know the inner products $\left\langle \psi\left(X^{(i)}\right), \psi\left(X^{(j)}\right)\right\rangle_{\mathcal{H}}$ for all $i, j \in [n]$ to perform linear regression in a Hilbert space. In addition, not even the map ψ is required, but rather the functional that yields the required inner products. Namely, we only need some function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ such that $K(x, \tilde{x}) = \langle \psi(x), \psi(\tilde{x})\rangle_{\mathcal{H}}$. This is formalized by the notion of a kernel:

Definition 2.8 (Kernel)**.** Given a nonempty set $\mathcal{X}$, a kernel is a symmetric continuous function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

Note that specifying K to have this inner product form imposes the constraint $K(x, x) \geq 0$. Consequently, we will focus on kernels that adhere to the positive semi-definite constraint, as defined below:

Definition 2.9 (Positive semi-definite kernel)**.** Given nonempty set $\mathcal{X}$, a kernel function $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ is positive semi-definite iff for any $\left\{X^{(i)}\right\}_{i=1}^{n} \subset \mathcal{X}$ and for any $\{c_i\}_{i=1}^{n} \subset \mathbb{R}$,

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j K\left(X^{(i)}, X^{(j)}\right) \geq 0.$$

Let us introduce the Kernel Regression framework by simplifying Equation (2.8) given the kernel function notation:

Theorem 2.10 (Kernel Regression)**.** *Let $\mathcal{H}$ be a Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Let $\psi : \mathbb{R}^d \to \mathcal{H}$ and let $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a kernel function such that $K(x, \tilde{x}) = \langle \psi(x), \psi(\tilde{x})\rangle_{\mathcal{H}}$. The minimum $\mathcal{H}$-norm minimizer of the loss:*

$$\mathcal{L}(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(y^{(i)} - \left\langle \theta, \psi\left(X^{(i)}\right)\right\rangle_{\mathcal{H}}\right)^2$$

is given by $\tilde{\theta} = \sum_{i=1}^{n} \left[y\hat{K}^{\dagger}\right]_i \psi(X^{(i)})$*, where* $\hat{K} \in \mathbb{R}^{n\times n}$ *is the positive semi-definite matrix with entries* $\hat{K}_{i,j} = K(X^{(i)}, X^{(j)})$*. Moreover, the corresponding predictor,* $\hat{f} : \mathbb{R}^d \to \mathbb{R}$*, is given by*

$$\hat{f}(x) = y\hat{K}^{\dagger}\hat{K}(X, x)$$

where $\hat{K}(X, x) \in \mathbb{R}^n$ *with entries* $\hat{K}(X, x)_i = K\left(X^{(i)}, x\right)$*.*

Proof. From the Representer Theorem 2.7, we already know $\tilde{\theta}$ has the form $\tilde{\theta} = \sum_{i=1}^{n} \alpha_i \psi\left(X^{(i)}\right)$. Thus, it only remains to show that the vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n) \in \mathbb{R}^{n\times 1}$ equals $y\hat{K}^{\dagger}$. In particular, from Equation (2.8), we know that we can rewrite our minimizer in matrix form as

$$\mathcal{L}(\alpha) = \frac{1}{2}\|y - \alpha\hat{K}\|_2^2$$

Now minimizing $\mathcal{L}(\alpha)$ is equivalent to solving the linear system $y = \alpha\hat{K}$, which by Theorem 2.4, the solution is given by $\alpha = y\hat{K}^{\dagger}$. Hence,

$$\tilde{\theta} = \sum_{i=1}^{n} \left[y\hat{K}^{\dagger}\right]_i \psi(X^{(i)})$$

Lastly, our predictor will be given by

$$\hat{f}(x) = \left\langle \tilde{\theta}, \psi(x) \right\rangle_{\mathcal{H}} = \left\langle \sum_{i=1}^{n} \left[y\hat{K}^{\dagger}\right]_i \psi\left(X^{(i)}\right), \psi(x) \right\rangle_{\mathcal{H}} = y\hat{K}^{\dagger}\hat{K}(X, x),$$

as we wanted to show. □

Notice that Theorem 2.10 show us that kernel regression method solves a linear regression with a square matrix, making it faster to linear solvers. Moreover, finding the minimum norm solution with kernel regression involves solving a convex optimization problem. With this approach, the optimization becomes way more simpler. Moreover, kernel regression offers interpretability of learned solutions in the sense that every prediction on a new sample is just a weighted linear combination of labels for training examples. Hence, it is possible to know which training examples were most influential in the prediction for a new sample.

2.3 Decision Trees, Gradient Boosting and LightGBM

In the preceding sections, we explored the landscape of Linear and Kernel Regression, methods that are able to capture linear relationships and complex nonlinear patterns within data. However, datasets often exhibit intricate structures that are challenging for a single model to capture comprehensively. At this point, a very interesting supervised learning model comes into play: *Gradient Boosting*. Before introducing it, let us introduce the decision trees, a key point to develop this algorithm.

2.3.1 Decision trees

Before dealing with decision trees, we need to present certain definitions and concepts from graph theory:

Definition 2.11 (Rooted Binary tree)**.** A rooted binary tree T is a tree data structure where each node has either 0 or 2 children. When an element in a binary tree has 2 children, we typically name them *left child* and *right child*.

Definition 2.12 (Node)**.** A node N in a binary tree is an element of the tree with the following properties:

- It contains a value or data, denoted as N_V.
- It may have a left child, denoted as N_{left}, and a right child, denoted as N_{right}. These children are also nodes in the binary tree.
- If a node has no left or right child, the corresponding child value is considered null.

Definition 2.13 (Parent, Root, Child, and Leaf Nodes)**.** For a node N in a binary tree:

- If N has a child, then N is the **parent** of that child.
- If N has no parent, then N is the **root** of the tree.
- Each of N_{left} and N_{right} is a **child** of N.
- If N has no children, then N is a **leaf** node, i.e., both N_{left} and N_{right} are null.

Now that we have defined these key concepts, we can introduce *Decision trees*. Decision trees provide us a highly effective solution to complex problems: they solve problems by breaking them down into a series of sequential and hierarchical decisions. These trees can be thought as a visual representation of a decision-making process, where each internal node, also known as a *decision or splitting node*, represents a decision based on a particular feature. Thus, each branch represents a potential outcome or state, and our data points are divided into one or the other based on the value of a specific feature. Finally, the leaves of the decision tree contain the final decisions made by the algorithm.

Decision tree construction

A decision tree makes these *splits* based on a set of rules that are determined through a process called recursive binary splitting. The goal of this process is to create what we call *binary splits* (two children at each node), that are able to partition the data into subsets, in a way that the model's predictive accuracy is improved. This procedure requires the selection of the best variable or feature, and the best threshold to split the data at each internal node of the tree. This is done iterating over all possible variables and splits. For quantitatively talk about best splits, we introduce impurity functions.

Definition 2.14 (Impurity Function)**.** An Impurity Function is a function Φ from the space of distributions of K classes (i.e. K positive reals summing to 1 that represent the probability of being in each class) to $\mathbb{R}$, with some additional properties:

- Φ is permutation invariant: it only depends on probabilities $p_1, ..., p_k$ but not in their order.
- Φ achieves its maximum when $p_i = 1/k$ for all i, i.e. when the distribution is uniform.
- Φ achieves its maximum when $p_i = 1$ for some i (and 0 for the rest), this is when the distribution concentrates in one class.

An impurity function has to be thought as a function that awards having a distribution with low variance or information. Some examples are:

- Entropy function: $\Phi = -\sum_{1 \leq i \leq K} p_i \log p_i$
- Miss-classification rate: $\Phi = 1 - \max_j p_j$
- Gini index: $\Phi = -\sum_{1 \leq i \leq K} p_i(1 - p_i)$

This is indeed what we want for our leaves: having only one class on it so we can be sure about their classification. Our strategy now will be to iteratively maximize the value of the impurity function in the children of each node. So, for each node, we will select the splitting that maximizes the sum of the impurity function on the children nodes, given the training data distribution. Finally, we will have a rule to stop the splitting. This rule can be having low amount of training examples on the given node or a high value of the impurity function, among others.

2.3.2 Gradient Boosting

Once the concept of decision tree has been introduced, we can now introduce *Gradient Boosting*, a supervised learning algorithm that we can describe as a potent ensemble learning technique. Let us begin introducing the definition of *weak learner*:

Definition 2.15 (Weak Learner)**.** A weak learner is a machine learning algorithm that performs slightly better than random chance. In the context of ensemble learning, weak learners are often used as building blocks to create a strong learner or a more robust model.

The main objective of Gradient Boosting is obtaining a strong classifier: it sequentially builds a series of weak learners, refining the model by compensating the errors made by the preceding models. In our implementation, we will consider these weak learners to be decision trees.

Thus, the final *strong learner* is an ensemble of all decision trees

$$H(X) = \alpha_1 h_1(X) + \alpha_2 h_2(X) + \cdots + \alpha_k h_k(X),$$

where X is our training dataset and $h_t(X)$ for $t \in \{1, \ldots, k\}$ are the weak learners (decision trees in our case).

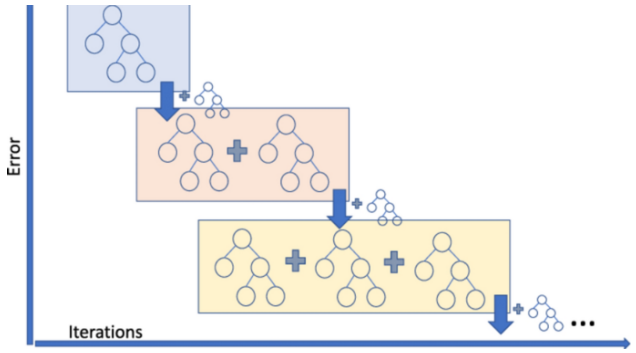


Figure 2.1: Gradient boosting diagram, source: [16]

The algorithm of gradient boosting follows the next idea:

- Before starting, we have a training dataset with m points $(X^{(i)}, y^{(i)})$, for $i \in \{1, \ldots, m\}$, with $X = (X^{(1)}, X^{(2)}, \ldots, X^{(m)})$ and $y = (y^{(1)}, y^{(2)}, \ldots, y^{(m)})$, a initial model $\hat{y} = F_0(X)$ and a loss function $\mathcal{L}(y, \hat{y}) > 0$.

  How can we improve our model $F_0(X)$? (since probably $F_0(X) \neq y$)

- Consider the residuals ε of our model such that $F_0(X) + \varepsilon = y$.

– Fit a new model $h_0(X)$ to ε so that the new prediction, $F(X) = F_0(X) + h_0(X)$, improve the previous one, $F_0(X)$. To do so, the new model $h_0(X, \theta)$ will be fitted to the *descend direction* of the gradient, that its

$$d = -\nabla\mathcal{L}(y, F(X)) = -\frac{\delta\mathcal{L}(y, F(x))}{\delta F(X)},$$

following the iterative method

$$F_{k+1} = F_k(X) + \lambda_k d_k,$$

where λ_k is the step-length.

Algorithm 1 Gradient Boosting Algorithm

Require: $X = \left(X^{(1)^T}, \ldots, X^{(m)^T}\right)^T, y = (y_1, \ldots, y_m)^T$, $h(X, \theta), \mathcal{L}(y, F(X))$ and $K \in \mathbb{N}$

1: $F_0(X) = \gamma$, where γ is a constant.
2: **for** $k = 1$ to K **do**
3: Compute $-\nabla\mathcal{L}\left(y, F_k(X)\right)$ ▷ (opposite to gradient).
4: Fit a model $h\left(X, \theta^k\right)$ to $-\nabla\mathcal{L}\left(y, F_k(X)\right)$.
5: Calculate the optimal step-length, λ_k by solving:

$$\lambda_k = \arg\min_{\lambda} \mathcal{L}\left(y, F_{k-1}(X) + \lambda h\left(X, \theta^k\right)\right)$$

6: Update $F_{k+1}(X) = F_k(X) + \lambda_k h\left(X, \theta^k\right)$.
7: **end for**

2.3.3 LightGBM

To fit our model, we first need to discern between to type of variables: categorical and numerical. Categorical variables are characterized by different categories or labels, such as gender or types of cars, while numerical variables are defined by measurable quantities, like height or temperature.

Thus, since several of our features are categorical, we have chosen LightGBM as a gradient boosting framework, which is well known for easily supporting categorical variables, making it a good candidate for our needs.

LightGBM [17] implements a conventional Gradient Boosting Decision Tree, where the trees grow leaf-wise; that is, given a condition, only a single leaf is split, depending on the gain, as we can see in Figure 2.2. Moreover this algorithm uses two novel techniques in order to reduce complexity by down-sampling the data and the features: Gradient-Based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB).

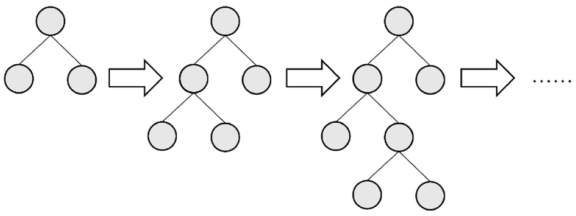


Figure 2.2: Leaf-wise tree growth, source [18]

The first one, GOSS technique, has the goal of selecting a subset of the data to use when training a gradient boosting model. It works by first sorting the training data by the gradients of the loss function with respect to the current model, and then selecting certain percentage of the data with the largest gradients, while randomly sampling the remaining instances. This allows the model to focus on instances that are more important for training, while still maintaining a diverse training set.

On the other hand, EFB aims to reduce the number of features used by regrouping mutually exclusive features (the ones that do not take non-zero values simultaneously) into bundles, treating them as a single feature. As proven in [17], finding the optimal bundle of exclusive features is NP-hard, but a greedy algorithm can achieve a good approximation ratio.

Chapter 3

Results

In this chapter we will present the results of the theory presented in Chapter 2. The implemented code can be found at https://github.com/silviag08/Flight-Fuel-Models.

First of all, we will consider a set of variables, renamed from table 1.1 for simplicity during coding, or a variation of them with the corrections presented in Section 1.

Column Name	Description
`Origin_Airport`	Origin Airport
`Destination_Airport`	Destination Airport
`model`	Aircraft type, using the ICAO designation
`seats`	Number of available seats
`dist`	Traveled distance, in kilometers, using the corresponding GCD and corrections described in subsection 1.2.1.
`depcount`	Number of times that the flight took place during the year 2017
`fuel_burn_total`	Grams of fuel burn during the flight per available seats and kilometer (g/ASK), obtained from IMPACT, the web-based modelling platform from EUROCONTROL, adding the correction of taxi fuel consumption described in subsection 1.2.2

Table 3.1: Column description of the dataset

Note that during training, we will use the variable `dist`, containing the corrections described in 1.2.1. Nevertheless, for a real application, corrections should be skipped since the model is already trained considering them.

Dataset	Number of Samples	Percentage
Total	58125	100%
Train	46500	80%
Test	11625	20%

Table 3.2: Split sets information

In order to assess our models accurately, so that we can gauge their performance on unseen data, we will split our data into two distinct groups, *train* and *test*, described in Table 3.2, that will use for training and predicting, respectively.

When evaluating a model, relying only on the training data can paint an overly positive picture of its capabilities: the model becomes too focused on the training data, to the point where it picks up on random variations or noise instead of the underlying patterns. While this may result in impressive performance on the train set, the model may struggle to effectively apply its learning's to new, unfamiliar data. When this happens, we say that the model is *overfitting*. By incorporating a separate test set, we can ensure that our assessment truly reflects the model's ability to handle new data and avoids the dangers of overfitting, giving us a more dependable and accurate measure of its performance.

To evaluate the performance of our models and to be able to compare them, setting a fixed metric is required. From [9], MAPE metric could have been a great option:

Definition 3.1 (MAPE)**.** Consider $\{Y\}_{i=1}^n$ and $\{\hat{Y}\}_{i=1}^n$ to be the sets of actual and predicted values respectively. Then, the mean absolute percentage error (MAPE) is defined as follows

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i - \hat{Y}_i|}{Y_i}$$

However, after examining the distribution of our data, we decided that a weighted metric could be used, since some of our flights routes were significantly more frequent than others, and hence they will also appear more while predicting. However, frequent flights are usually shorter, and so is their fuel consumption. To tackle this problem, we consider a new set of weights

$$\{W_i\}_{i=1}^n = \frac{\texttt{depcount}_i \cdot \texttt{fuel\_consumption}_i}{\sum_{j=1}^n \texttt{depcount}_j \cdot \texttt{fuel\_consumption}_j},$$

and thus, a new metric,

Definition 3.2 (WMAPE)**.** Consider $\{Y_i\}_{i=1}^n$ and $\{\hat{Y}_i\}_{i=1}^n$ to be the sets of actual and predicted values, respectively. Consider $\{W_i\}_{i=1}^n$ to be the set of weights associated with a respective Y_i. We usually assume that $\sum_{i=1}^n W_i = 1$. Then, we define the weighted mean absolute percentage error (WMAPE) as follows

$$\text{WMAPE} = \sum_{i=1}^{n} W_i \cdot \frac{|Y_i - \hat{Y}_i|}{Y_i}.$$

Note that this metric assigns greater significance to flights that are either highly frequent or have substantial fuel consumption, i.e, to flights that contribute a larger proportion to the overall fuel consumption. If we had considered an unweighted metric, it had could given too much importance to very rare or/and short flights, which, in reality, contribute minimally

to the total global fuel consumption. In practical applications, those are not from big interest.

Throughout this section, we will use the WMAPE metric to evaluate our models, although we will refer to it as MAPE for simplicity.

3.1 Linear Regression

In this section we will present the results following the linear regression approach presented in subsection 2.1. From the paper *Analytical Models for CO_2 Emissions and Travel Time for Short-to-Medium-Haul Flights Considering Available Seats* [9], we know that our baseline is considering a linear regression with the following features:

Column Name[1]	**Description**
`seats`	Number of available seats
`dist`	Traveled distance, in kilometers, using the corresponding GCD and corrections described in subsection 1.2.1.
`inv_dist`[2]	Inverse of `dist`
`fuel_burn_total`	Grams of fuel burn during the flight per available seats and kilometer (g/ASK), obtained from IMPACT the web-based modelling platform from EUROCONTROL, adding the correction of taxi fuel consumption described in subsection 1.2.2

Table 3.3: Column description of the dataset

In order to apply classical statistical results to evaluate how good our model is, for instance, performing an analysis of variance using ANOVA, we need to check that the assumptions mentioned at the beginning of subsection 2.1 are acceptable for our model. That is, checking whether

$$\varepsilon^{(i)} = y^{(i)} - E(y_i \mid X_i) = y^{(i)} - \left(\tilde{\theta}_0 + \tilde{\theta}_1 X_1^{(i)} + \ldots + \tilde{\theta}_d X_d^{(i)}\right) \quad \text{for } i \in \{1, 2, \ldots, d\}$$

are close to being independent and identically distributed following a normal distribution and, in particular, whether the linearity, the constancy of variance and the normality assumptions hold. Note that one cannot check that directly because θ_i and, therefore, $\varepsilon^{(i)}$ are unknown.

[2]Note that `inv_dist` is considered since `fuel_burn_total` has units $\frac{grams}{seats \cdot km}$. The feature `inv_seats` was also considered but it decreased the accuracy of the model.

[1]Note that the column `depcount` is not present in table 3.3. This feature represents the frequency of a flight route completed in 2017. While it does not offer pertinent information for training our model, it is valuable for accounting for total fuel consumption.

Instead, what one can analyze is whether that is approximately the case by analyzing the best estimate we have for $\varepsilon^{(i)}$, which are the residuals:

$$e^{(i)} = y^{(i)} - \left(\theta_0 + \theta_1 X_1^{(i)} + \ldots + \theta_d X_d^{(i)}\right) = y^{(i)} - \hat{y}^{(i)} \quad \text{for } i \in \{1, 2, \ldots, d\} \tag{3.1}$$

Those are crucial for analyzing our model, since they "magnify" the lackings of it, and help discover ways to fix it. In order to proceed with this study, we will consider a graphical analysis of them, using the *normal probability plot of residuals*, also known as Q-Q plot. If the normality assumption holds, residuals should be placed more or less along the identity line [19].

Once our model is fitted, we obtain the following polynomial:

$$\texttt{fuel\_burn\_total} = 21.5839 + 8351.0606 \cdot \frac{1}{\texttt{dist}} + 0.0051 \cdot \texttt{dist} - 0.0806 \cdot \texttt{seats} \tag{3.2}$$

Taking now 100 samples of our data and plotting a Q-Q plot, we can appreciate in Figure 3.1 that the residuals do not follow a normal distribution (red line).

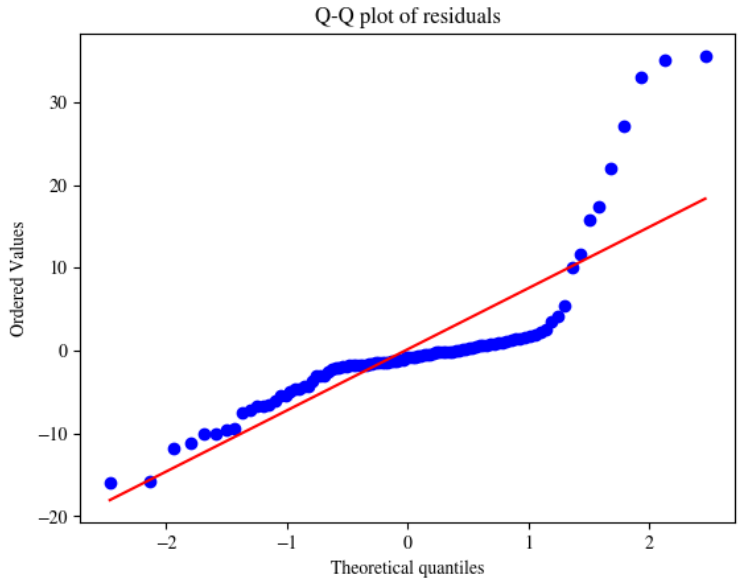


Figure 3.1: Q-Q plot of residuals taking 100 random samples

When this happens, the usual procedure is to apply a non-linear transformation to our features, taking into account their own distribution, in order to achieve normality on the residuals. However, in our case, after apply several classical non-linear transformation to tackle this problem, this did not work. Moreover, we do not have information to know which kind of transformation should we apply. This arises a problem: when not satisfying the normal lineal model assumptions, we cannot use theoretical results to improve it. This points

out that the lineal model is not the best one to adjust our data.

Yet, we can use our metric, defined in 3.2 to check the accuracy of the model considered in Equation (3.2). For our data, the error is very large, reaching 16.86% on the test set. This may be due to our data distribution, as shown in 1.1. To address this problem, we have also trained 4 different models, splitting our data by distance, as shown in Table 3.4, following the EUROCONTROL haul's division [20],

Haul Type	**Distance**	**N. of Samples**	**Samples train**	**Samples test**
Short Haul	`dist` < 500	8730	6984	1746
Medium Haul	$500 \leq$ `dist` < 1500	26392	21113	5279
Medium Long Haul	$1500 \leq$ `dist` < 4000	19601	15680	3921
Long	`dist` ≥ 4000	3402	2721	681

Table 3.4: Description of Haul Types based on Distance (`dist`), based on [20]

with the purpose of finding a model, even if multiple of them, that fits better our data. Thus, we will consider a different model that has been trained with only data belonging to one of the different hauls, having then 4 different models for our data. We will refer to this model as *Linear Regression by haul*. As we can appreciate in Table 3.5, the first table shows us that this methodology is not good for every set of data: for *Short* and *Medium* hauls, the accuracy decreases substantially, while for the others it increases notably, with a difference of a 12.808% of MAPE in the *Long* haul dataset.

With the purpose of increasing the accuracy, we considered adding new variables. In particular, we studied the addition the variable `CAT`, a variable used during the corrections applied to the distance in subsection 1.2.1. This variable divides the data into 6 different categories, based on the aircraft model (`model`). We introduced this variable as 6 different binary variables, each one representing one of the categories, taking values 1 if the sample belongs to a the category and 0 otherwise. As we can see reflected in the table 3.5, this implementation increases our results significantly.

Moreover, a new model was considered, adding in this case the variable `model` instead of `CAT`. Following the same approach as before, we had to add 91 binary variables, corresponding to each of the different models of our dataset. We expected the model to be overfitting, but as shown in last table of Table 3.5, the accuracy of the train and test sets is very similar. In particular, training different models for each type of haul (see *Linear Regression by haul+* `model`) gives us an outstanding performance compared to our first model.

Linear regression					
	Short	**Medium**	**Medium Long**	**Long**	**Total**
MAPE Train	-	-	-	-	17.754%
MAPE Test	19.721%	8.644%	19.432%	40.466%	16.860%
Linear regression by haul					
	Short	**Medium**	**Medium Long**	**Long**	**Total**
MAPE Train	20.805%	9.790%	11.117%	31.595%	17.754%
MAPE Test	20.636%	8.818%	11.468%	27.658%	16.860%

Linear Regression + `CAT`					
	Short	**Medium**	**Medium Long**	**Long**	**Total**
MAPE Train	-	-	-	-	14.988%
MAPE Test	16.480%	8.650%	11.582%	41.428%	14.659%
Linear Regression by haul + `CAT`					
	Short	**Medium**	**Medium Long**	**Long**	**Total**
MAPE Train	15.194%	9.250%	7.350%	31.281%	14.988%
MAPE Test	14.835%	8.115%	7.794%	28.178%	14.659%

Linear Regression + `model`					
	Short	**Medium**	**Medium Long**	**Long**	**Total**
MAPE Train	-	-	-	-	9.666%
MAPE Test	9.091%	5.580%	7.768%	32.216%	9.356%
Linear Regression by haul + `model`					
	Short	**Medium**	**Medium Long**	**Long**	**Total**
MAPE Train	7.852%	4.465%	5.030%	21.445%	9.666%
MAPE Test	7.794%	4.126%	5.147%	21.455%	9.356%

Table 3.5: Linear Regression accuracy using MAPE metric

3.2 Kernel Regression

In this section we will discuss the results from the theory presented in section 2.2. As developed, kernel regression tries to capture the underlying pattern or trend in the data without making strong assumptions about the form of the relationship between variables. In our case, our selected kernel function will be the Laplacian,

$$k(x, y) = \exp(-\gamma \|x - y\|_1),$$

where x, y are the input vectors, $\gamma \in \mathbb{R}$ and $\|x - y\|_1$ is the Manhattan distance between the input vectors. The selection of this kernel function was the result of systematically evaluate various kernels through a trial-and-error approach. The goal was to identify the kernel that yielded optimal performance based on the selected metric. Note that this kernel depends on γ, a parameter to be set. After some exploration, we decided to set $\gamma = 0.005$.

The main problem of this approach was that we have a huge set of data, and as developed in theory, the computation of an inverse matrix is required. However, after different implementation approaches and the use of at least 16GB of memory, were able to train and predict with this method. Nevertheless, we would like to remark on the computational time that this method requires.

As illustrated in Table 3.6, training a specific model for each haul does not give significant better results for every haul using kernel regression. Specifically, the only one for which applying a kernel regression by haul results in better accuracy is for the *Long* haul dataset.

Moreover, it is clear comparing Tables 3.5 and 3.6 that kernel regression performs generally better than linear regression. In particular, we cannot see an outstanding improvement using categorical variables `CAT` and `model` as we saw with linear regression, but that is because the model is already significantly better. However, note that *Kernel Regression by haul* + `model` exhibits worse results than *Kernel Regression* + `model`. In particular, this last one is also giving worst results than *Linear Regression by haul* + `model` when comparing most of the haul sets errors. Nevertheless, it reduces the error of the *Medium* haul set more than 1%, where most of our data belongs. This is the reason why this kernel model obtains the lower MAPE on the total test set until now.

Kernel Regression					
	Short	**Medium**	**Medium Long**	**Long**	**Total**
MAPE Train	-	-	-	-	9.223%
MAPE Test	10.155%	7.194%	4.768%	30.201%	9.850%
Kernel Regression by haul					
	Short	**Medium**	**Medium Long**	**Long**	**Total**
MAPE Train	10.077%	7.437%	4.256%	24.675%	9.223%
MAPE Test	9.614%	7.432%	4.959%	26.588%	9.850%

Kernel regression + `CAT`					
	Short	**Medium**	**Medium Long**	**Long**	**Total**
MAPE Train	-	-	-	-	9.051%
MAPE Test	10.116%	7.120%	4.529%	29.802%	9.741%
Kernel regression by haul + `CAT`					
	Short	**Medium**	**Medium Long**	**Long**	**Total**
MAPE Train	9.923%	7.311%	3.986%	24.415%	9.051%
MAPE Test	9.495%	7.305%	4.694%	26.438%	9.741%

Kernel Regression + `model`					
	Short	**Medium**	**Medium Long**	**Long**	**Total**
MAPE Train	-	-	-	-	7.914%
MAPE Test	8.631%	5.976%	3.920%	28.073%	8.492%
Kernel Regression by haul + `model`					
	Short	**Medium**	**Medium Long**	**Long**	**Total**
MAPE Train	8.670%	6.462%	3.531%	22.786%	7.914%
MAPE Test	8.157%	6.405%	4.114%	24.868%	8.492%

Table 3.6: Kernel regression error using MAPE metric

3.3 LightGBM

In this last section, we will present the results of the model that has given better results: LightGBM. As commented in subsection 2.3.3, this model is profitable for regression tasks such ours, due to its efficient and scalable gradient boosting framework, optimized for large datasets, and its ability to handle complex non-linear relationships.

In this case, after considering the results obtained in sections 3.1 and 3.2, we can state that the feature `model` holds significant importance. Therefore, the presented results include directly this feature.

Like any machine learning model, LightGBM includes adjustable parameters that influence its performance, called *hyperparameters*. Finding the best hyperparameters for our model based on the data, a process called *finetuning*, is an extremely important procedure in order to obtain a good accuracy. The specific parameters we have set include:

- `objective` (`'regression'`): loss function to be selected. In this case, `'regression'`, was chosen, corresponding to the *Mean Squared Error*:

$$\text{Mean Squared Error} = \frac{1}{n}\sum_{i=n}^{n}(Y_i - \hat{Y}_i)^2.$$

- `learning rate (0.15)`: step size at each iteration while moving toward a minimum of the loss function.
- `max_depth (8)`: maximum depth of a tree. Can be used to prevent overfitting.

Other hyperparameters were left with default values. In particular, for training this model we have considered another dataset, called *validation*.

Dataset	Number of Samples	Percentage
Total	58125	100%
Train	46500	80%
Validation	5813	10%
Test	5812	10%

Table 3.7: Split sets information

The monitoring of performance is specifically conducted on this validation dataset, using *early stopping*:

Definition 3.3 (Early Stopping)**.** Early stopping technique monitors the performance metrics on the validation set at regular intervals. If the performance on the validation set stops improving or starts deteriorating, the training process is halted before completing all the planned iterations.

Specifically, we set the number of iterations for early stopping to 15. With the specified parameters parameters, we obtained the following results:

Linear Regression + `model`					
	Short	**Medium**	**Medium Long**	**Long**	**Total**
MAPE Train	-	-	-	-	9.666%
MAPE Test	9.091%	5.580%	7.768%	32.216%	9.356%
Kernel Regression + `model`					
	Short	**Medium**	**Medium Long**	**Long**	**Total**
MAPE Train	-	-	-	-	7.914%
MAPE Test	8.631%	5.976%	3.920%	28.073%	8.492%
LightGBM					
	Short	**Medium**	**Medium Long**	**Long**	**Total**
MAPE Train	-	-	-	-	2.188%
MAPE Validation	-	-	-	-	3.110%
MAPE Test	3.873%	2.629%	1.532%	6.790%	3.460%

Table 3.8: Comparison table of different models using MAPE metric

As illustrated in Table 3.8, this model gives outstanding results compared to previous approaches, reaching an error of 3.46% on the test set.

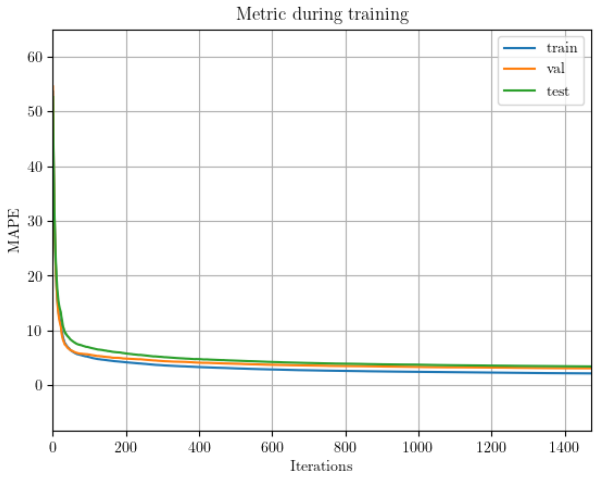


Figure 3.2: Training steps vs MAPE

One thing worth noticing about this model is illustrated in Figure 3.2, depicting the evolution of the metric during the training steps of the model. We can see a huge improvement in the first 100 steps (as one could expect) and a reasonable improvement in the next 1000 steps. The remark is that all train, validation and test sets have a similar learning curve, so we do not detect overfitting.

Chapter 4

Application: comparing rail and flight CO_2 emissions

In this chapter we present a real-world application of our LightGBM model 3.3, the model developed with better accuracy. The goal of this application will be analyzing the reduction of CO_2 emissions if a certain flight route will be replaced by a rail one, considering AVE (high-speed train) rail routes. In particular, we will present results regarding a week of air traffic in May 2023, with flights arriving/departing within Spain from the 1^{st} to the 7^{th} of May 2023 [21, 22]. Note that some flights might depart on the 30^{th} April or land on the 8^{th} of May. Further, the 1^{st} of May is a public holiday in Spain, and the 2^{nd} May is a public holiday in the region of Madrid. This impacts directly the flight traffic. Information regarding CO_2 emissions of AVE routes is obtained from *EcoPassenger* calculator [23].

This dataset comprises 1538 flights which, in contrast to the previous data used to train 1.2, they are not unique and do not include a `depcount` column. In other words, each row in our dataset represents an individual flight occurrence. The flight distance in this case is found within a more restricted range, varying between 277 km and 894 km. Flight distance distribution in Figure 4.1 illustrates that most part of the flights are classified as *Short Haul* flights, while the others would belong to the *Medium Haul* group, the description of which is made in Table 4.1. Therefore, following the results obtained in Table 3.8, we can expect an error between 2.5% and 4% if this new data behaves similary as our LightGBM test dataset.

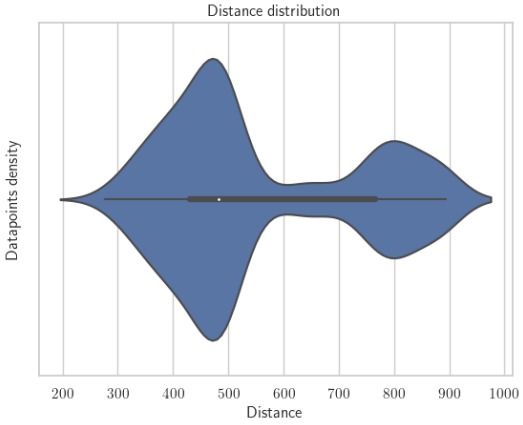


Figure 4.1: Violin plot of the distance distribution

4.1 Data preprocessing

For simplicity, we transformed the data into a dataset containing unique routes of flights, and created a new column named `depcount`, representing the number of times a route has taken place, as we had before. With this change, we obtain 226 rows of unique flight routes. In particular, data follows the distribution illustrated in Table 4.1.

Haul Type	Distance	N. of Samples	Expected MAPE
Short Haul	`dist` < 500	101	3.873%
Medium Haul	$500 \leq$ `dist` < 1500	125	2.629%

Table 4.1: Description of Haul Types based on Distance (`dist`), based on [20]

The dataset used for this application contains models that our LightGBM model has not seen before, since they were not in the database used to train the model. To tackle this problem, we decided to substitute them by the more similar ones in our database, as shown in Table 4.2. With this change, we ensure a better accuracy for the prediction of the fuel consumption of these flights.

Model to be replaced	New Model	N. of flights affected[3]
AT75	AT72	16
AT76	AT72	14
A20N	A320	135
A21N	A321	14
B38M	B737	18

Table 4.2: Model changes

The provided dataset contained 16 flights[4] without `model`, and thus, without number of available seats (`seats`). Even if LightGBM is able to handle missing values, predictions of these flights were not good, returning negative values. However, since `dist`, `Arrival_Airport` and `Departure_Airport` were provided, we decided to map these flights with the mean of `seats` of flights with the same flight route. In particular, predictions of these 16 flights, corresponding to 6 different flight routes, were also computed taking the mean value of the predictions for those flights with the same flight route.

[3]This column indicates the number of flights affected, not the flight routes affected.
[4]Again, we are talking about different flights, not flight routes.

4.2 Results

For this study, we have chosen three distinct flight routes to analyze their respective CO_2 emissions and draw comparisons with the rail emissions for the same routes. The selection criteria included routes with no layovers and rail travel time of less than 3 hours.

Focusing on the capital of the country, Madrid, which boasts superior rail connectivity, we considered in Figure 4.2 a description of Madrid arrivals and departures. It is evident that the flight routes between Madrid-Barcelona or Barcelona-Madrid (typically featuring an equal number of flights due to their round-trip nature), are the most frequent ones, with a number of flights performed of 126 and 128 respectively. Moreover, it represents one of the routes with the most efficient AVE connectivity.

Figure 4.2: Madrid Arrivals and Departures

For our study, we have also chosen Madrid-Málaga and Madrid-Sevilla routes, along with their corresponding roundtrips. Other routes may not have AVE direct connection with Madrid, but could also be considered. However, we will focus on the mentioned ones. From now on, when we mention Madrid-Barcelona, Madrid-Málaga, or Madrid-Sevilla, we will also be encompassing their including round-trip journeys in our discussions.

For these routes, we find in Table 4.3 the number of flights with destination or origin Madrid, representing together the 27,2% of our dataset. It also includes information obtained from [23] about the train model, travel time and CO_2 emissions per passenger that will be used in our study. Note that the PAX in this table stands for passenger.

Route	N. of flights
Madrid-Barcelona	254
Madrid-Málaga	97
Madrid-Sevilla	68

Train model	CO_2 kg/PAX	Duration
AVL 6303	18.4	2:45 h
AVL 2216	13.4	2:56 h
AVE 2122	15.2	2:55 h

Table 4.3: Information regarding number of flights and train models selected, from or to Madrid.

As we have seen in previous sections, fuel consumption varies in function of the aircraft model. In Table 4.4, we present prediction results in function of the model, located in column *Fuel Pred (g/ASK)*. An additional column required for our analysis is also computed: *CO_2 (t/flight)*. These can be easily estimated from fuel burn using $\mathrm{CO_2}$ (g) $= 3.157 \cdot$fuel burn (g) , given by *The International Air Transport Association (IATA)* [24]. However, in order to make a fear comparison with EcoPassenger's train emissions, we need to use a larger factor,

$$\mathrm{CO_2}\ (\mathrm{g}) = 3.78 \cdot \text{fuel burn (g)}\ [25]. \tag{4.1}$$

This is due to its methodology approach, where they consider not only emissions during the travel, but also during all life cycle of the respective energy source. With this information, we are able to compute the column *CO_2 (t/flight)* using a simple conversion factor:

$$\mathrm{CO_2}\ (\mathrm{t/flight}) = 3.78 \cdot 10^{-6} \cdot \text{ Fuel Pred. (g/ASK)} \cdot \texttt{dist} \cdot \text{seats},$$

leading to tonnes of $\mathrm{CO_2}$ per flight. Lastly, we have *Total CO_2 (t)*, representing the tonnes of $\mathrm{CO_2}$ emitted by all flights by model, so multiplying *CO_2 (t/flight)*· `depcount`.

At the end of each table, we can observe the *Total CO_2 Emissions (t)*, which is the sum of the last column and represents the total emissions during a week for each of the routes, including both ways.

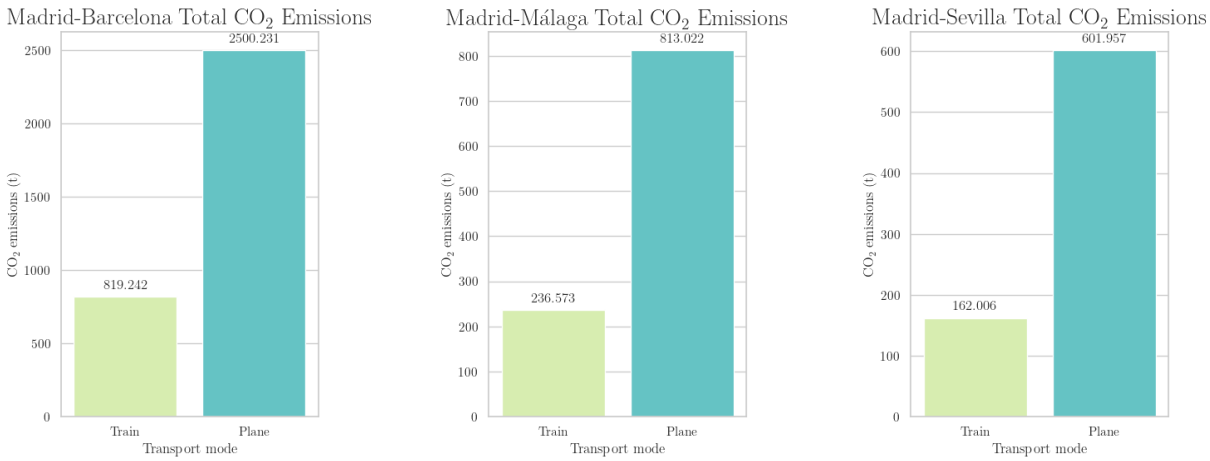


Figure 4.3: Comparison total flight and train $\mathrm{CO_2}$ emissions.

In Figure 4.3 we illustrate the difference between the rail and flight emissions for each one of the routes, considering the total $\mathrm{CO_2}$ emissions per route. As we can appreciate, the reductions are outstanding, decreasing approximately a 70% the $\mathrm{CO_2}$ emissions for every route. In Table 4.5 we can see a better description of the obtained values.

Madrid-Barcelona Flight Route, 483 km					
Model	**Seats**	`depcount`	**Fuel Pred. (g/ASK)**	**CO_2 (t/flight)**	**Total CO_2 (t)**
A21N	183	48	25.726	8.597	412.644
A319	124	18	37.631	8.521	153.372
A320	150	126	31.233	8.555	1060.829
A321	185	28	28.424	9.602	268.856
B788	242	10	37.083	16.387	163.869
B789	335	18	33.176	20.295	365.307
CRJX	81	2	32.518	4.810	9.620
-	186	6	32.256	10.956	65.733
Total CO_2 Emissions (t)					**2500.231**

Madrid-Málaga Flight Route, 430 km					
Model	**Seats**	`depcount`	**Fuel Pred. (g/ASK)**	**CO_2 (t/flight)**	**Total CO_2 (t)**
A320	150	36	32.483	7.922	285.181
A321	232	6	32.455	12.242	73.451
B738	160	48	31.725	8.253	396.126
CRJX	81	1	35.509	4.676	4.676
-	156	3	33.043	8.381	25.142
-	181	3	32.220	9.482	28.445
Total CO_2 Emissions (t)					**813.022**

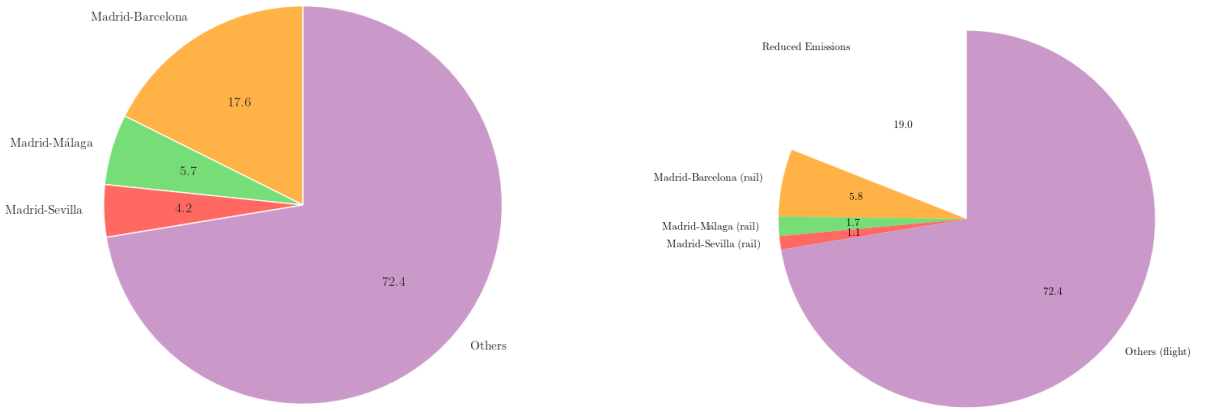
Madrid-Sevilla Flight Route, 395 km					
Model	**Seats**	`depcount`	**Fuel Pred. (g/ASK)**	**CO_2 (t/flight)**	**Total CO_2 (t)**
A320	150	36	33.621	7.529	271.060
A321	185	16	29.510	8.151	130.410
A321	232	8	33.117	11.471	91.766
A359	348	4	40.761	21.178	84.712
B738	160	2	32.175	7.686	15.372
CRJX	81	2	35.706	4.318	8.636
Total CO_2 Emissions (t)					**601.957**

Table 4.4: Predictions for flights between Madrid-Barcelona, Madrid-Málaga and Madrid-Sevilla (both ways)

Route	**Flight CO_2**	**Rail CO_2**	**CO_2 reduction**	**Reduction %**
Madrid-Barcelona	2500.231	819.242	1680.989	67.2%
Madrid-Málaga	813.022	235.433	577.589	71%
Madrid-Sevilla	601.957	162.006	439.951	73%

Table 4.5: Comparative results between flight and rail CO_2 emissions, expressed in tonnes.

Recall that our data provides information regarding a week of airtraffic within Spain. Thus, we can also draw conclusions about weekly CO_2 emissions. Using our LightGBM model 3.3 and Equation (4.1), we estimate the total weekly CO_2 emissions to be 14186 tonnes. Through our analysis, we explore the potential impact of replacing all the studied flight routes, namely Madrid-Barcelona, Madrid-Málaga, and Madrid-Sevilla (round trips), with their corresponding rail routes, assuming that all passengers will choose a rail alternative. This substitution is projected to reduce the total weekly CO_2 emissions by 19%, equating to a significant prevention of 2697 tonnes of CO_2, illustrated in Figure 4.4. Note that this situation is ideal, as not all flights on a certain route can be replaced, and not all passengers on those flights may choose a rail alternative.



(a) Emissions produced by flights only

(b) Emissions with train alternatives in selected routes

Figure 4.4: Weekly emission of airtraffic within Spain in (a) and emissions if replacing the selected routes by train alternatives in (b)

Chapter 5

Conclusion

In this thesis we have developed the theory behind the predictive models presented: Linear Regression, Kernel Regression and LightGBM. As anticipated, due to its complexity compared to the others, the LightGBM model yields the most accurate results in predicting fuel consumption, boasting a remarkably low MAPE of 3%. In contrast with other fuel consumption calculators, this model has the peculiarity of not requiring extensive data on the flight route for accurate predictions, relying only the aircraft model, the travel distance and the number of available seats. Moreover, our model is a more dynamic tool compared to alternative calculators, enabling the simultaneous computation of fuel consumption for multiple flight routes.

Furthermore, we have applied noteworthy corrections to our data, specifically addressing flight distance and taxi fuel consumption. In the case of taxi fuel consumption, we tried to establish accurate approximations for 91 different aircraft models, aiming to enhance the precision of our analyses.

Based on the results from our application study, which compares CO_2 emissions between rail and flight travel, we can see the importance of considering other less polluting travel options when feasible. The significant reduction in emissions underscores the importance of embracing more environmentally-friendly travel alternatives.

As future work, we could perform a further exploration to try to find better parameters for our LightGBM model. Also, if the data is available, we could study the inclusion of additional features that may impact fuel consumption, such as weather conditions, air traffic, or fuel prices. Even if this adds complexity to our model, the incorporation of these features may enhance notably the accuracy of it. Another point of improvement could be performing an error study, where we take into account all our assumptions and data-preprocessing to give a more robust value of the error.

Bibliography

[1] SESAR Joint Undertaking. European ATM Master Plan—Executive View. 2020. Available at https://www.sesarju.eu/masterplan2020.

[2] Joint Technical Programme. Clean Sky 2. 2015. Available at https://ec.europa.eu/research/participants/data/ref/h2020/other/guide-appl/jti/h2020-guide-techprog-cleansky-ju_en.pdf.

[3] DLR. DEPA 2050—Development Pathways for Aviation up to 2050. *Technical Report, German Aerospace Center*, 2020. Available at https://elib.dlr.de/142185/1/DEPA2050_StudyReport.pdf.

[4] European Commission. Regulation (EU) 2017/2392 of the European Parliament and of the Council of 13 December 2017 Amending Directive 2003/87/EC to Continue Current Limitations of Scope for Aviation Activities and to Prepare to Implement a Global Market-Based Measure from 2021. 2017. Available at www.legislation.gov.uk/eur/2017/2392/adopted/data.pdf.

[5] European Commission. Resolution A40-19: Consolidated Statement of Continuing ICAO Policies and Practices Related to Environmental Protection—Carbon Offsetting and Reduction Scheme for International Aviation (CORSIA). 2019. Available at www.legislation.gov.uk/eur/2017/2392/adopted/data.pdf.

[6] Europa Press Turismo. PSOE y Sumar reducirán los vuelos domésticos en las rutas con alternativa ferroviaria de hasta 2,5 horas. *Europa Press*. October 24, 2023. Available at https://www.europapress.es/turismo/nacional/noticia-psoe-sumar-reduciran-vuelos-domesticos-rutas-alternativa-ferroviara-25-horas-20231024142141.html.

[7] Air France. Combined plane + train tickets. Available at https://wwws.airfrance.es/en/information/prepare/voyages-combines-avion-train.

[8] ICAO. Carbon Emissions Calculator (ICEC). Available at https://www.icao.int/environmental-protection/Carbonoffset/Pages/default.aspx.

[9] Adeline Montlaur, Luis Delgado, and César Trapote-Barreira. Analytical Models for CO_2 Emissions and Travel Time for Short-to-Medium-Haul Flights Considering Available Seats. *Sustainability*, 13(18), 2021.

[10] ICAO. Carbon Emissions Calculator Methodology. Version 11. 2018. Available at www.icao.int/environmental-protection/CarbonOffset/Documents/Methodology%20ICAO%20Carbon%20Calculator_v11-2018.pdf.

[11] Performance Review Comission. Performance Review Report 2022. 2023. Available at https://www.eurocontrol.int/publication/performance-review-report-prr-2022.

[12] Wikipedia. Taxiing. Available at https://en.wikipedia.org/wiki/Taxiing.

[13] EUROCONTROL. Taxi times - Winter 2019-2020. Available at https://www.eurocontrol.int/publication/taxi-times-winter-2019-2020.

[14] RECAT-EU. European Wake Turbulence Categorisation and Separation Minima on Approach and Departure. 2018. Available at https://www.eurocontrol.int/sites/default/files/2021-07/recat-eu-released-september-2018.pdf.

[15] Adityanarayanan Radhakrishnan. Modern Machine Learning: Simple Methods that Work. Available at https://web.mit.edu/modernml/course/.

[16] Iniyan and Jebakumar. Mutual Information Feature Selection (MIFS) Based Crop Yield Prediction on Corn and Soybean Crops Using Multilayer Stacked Ensemble Regression (MSER). *Wireless Personal Communications*. June of 2021. Available at https://link.springer.com/article/10.1007/s11277-021-08712-9.

[17] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Neurips*, 30, 2017.

[18] Bv Wang, Zhibin Zhang, Xiao Wang, Shuai Zhao, Zao Yi, and Shunguang Hu. Object-Based Mapping of Gullies Using Optical Images: A Case Study in the Black Soil Region, Northeast of China. *Remote Sensing*, 12:487, 02 2020.

[19] Josep Ginebra. Course notes, Normal Linear Model.

[20] EUROCONTROL. Aviation Sustainability Unit, Think Paper 11 - 3 June 2021. Available at https://www.eurocontrol.int/sites/default/files/2021-06/eurocontrol-think-paper-11-plane-and-train-right-balance.pdf.

[21] Schäfer Matthias, Strohmeier Martin, Lenders Vincent, Martinovic Ivan, and Wilhem Matthias. "Bringing up OpenSky: A Large-Scale ADS-B Sensor Network for Research". *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, pages 83–94, 2014. Available at https://opensky-network.org/data/impala.

[22] OpenSky. A Quick Guide To OpenSky's Impala Shell. Available at https://opensky-network.org/data/impala.

[23] EcoPassenger. Compare the energy consumption, the CO_2 emissions and other environmental impacts for planes, cars and trains in passenger transport. Available at https://ecopassenger.hafas.de.

[24] IATA. IATA Carbon Offset Program—Frequently Asked Questions. version 10.1. 2020. Available at https://www.iata.org/contentassets/922ebc4cbcd24c4d9fd55933e7070947/icop_faq_general-for-airline-participants.pdf.

[25] EcoPassenger. Environmental Methodology and Data. 2016. Available at https://ecopassenger.hafas.de/hafas-res/download/Ecopassenger_Methodology_Data.pdf.

[26] Harshad Khadilkar and Hamsa Balakrishnan. Estimation of Aircraft Taxi-out Fuel Burn using Flight Data Recorder Archives. *Massachusetts Institute of Technology, Cambridge, MA 02139, USA*. Available at https://www.mit.edu/~hamsa/pubs/KhadilkarBalakrishnanGNC2011.pdf.

[27] StackExchange Aviation. How much fuel does the A380 use for taxiing before takeoff? . *Genx Engine*. December 19, 2023. Available at https://aviation.stackexchange.com/questions/50668/how-much-fuel-does-the-a380-use-for-taxiing-before-takeoff.

[28] Michael Gebicki. How much fuel do aircraft burn when they taxi? *The Sydney Morning Herald*. February 9, 2018. Available at https://www.smh.com.au/traveller/reviews-and-advice/how-much-fuel-do-aircraft-burn-when-they-taxi-20180209-h0vtp4.html.

[29] Jay Stowe. Flip the Script: GEnx Program Caps Off Big Year With LATAM Airlines Deal . *Genx Engine*. December 19, 2023. Available at https://www.ge.com/news/taxonomy/term/8570.

[30] Airline Pilot Central. EMB 145 Fuel Savings . January 4, 2019. Available at https://www.airlinepilotcentral.com/articles/news/emb-145-fuel-savings.html.

[31] Aviaddicts. Fuel Flows . Available at https://forums.aviaddicts.com/wiki/print.php?page=ejet:fuel_consumption.

[32] Abdulrazaq Lemu Salihu, Shannon M. Lloyd, and Ali Akgunduz. Electrification of airport taxiway operations: A simulation framework for analyzing congestion and cost. *Transportation Research Part D: Transport and Environment*, 97:102962, 2021.

[33] LET. L410 Flight Manual . April, 2019. Available at https://x-plane.hu/L-410/download/L410%20Flight%20Manual.pdf.

Appendix A

Taxi fuel consumption by aircraft model

This appendix contains the table comprising the taxi fuel consumption per aircraft model used in our computations. The column *Available Fuel* corresponds to the taxi fuel consumption obtained from column *Source*. When it is informed, it serves directly as *Taxi Fuel*, the column from which we extract the taxi fuel consumption. When not informed, the mean of aircraft models with known *Available Fuel* for each *CAT* category is employed. CAT categories are obtained using RECAT-EU classification [14].

Model	CAT	Available Fuel (kg/s)	Taxi Fuel (kg/s)	Source
A140	E		0.1797	
A148	E		0.1797	
A306	C		0.2775	
A310	C		0.2775	
A318	D		0.2088	[26]
A319	D	0.2088	0.2088	[26]
A320	D	0.2123	0.2123	[26]
A321	D	0.2208	0.2208	[26]
A332	B	0.3287	0.3287	[26]
A333	B	0.3287	0.3287	[26]
A340	B		0.6670	
A343	B	0.6350	0.6350	[26]
A345	B	0.6350	0.6350	[26]
A346	B	0.6350	0.6350	[26]
A359	A		0.6670	
A388	E	0.7370	0.7370	[27]
AN24	E		0.1797	
AN26	F		0.1797	
AN38	E		0.3330	
AT43	E		0.1797	
AT46	E		0.1797	
AT72	E		0.1797	

Model	CAT	Available Fuel (kg/s)	Taxi Fuel (kg/s)	Source
ATP	E		0.1797	
B190	F		0.3330	
B350	F		0.3330	
B462	E		0.1797	
B463	E		0.1797	
B712	E		0.1797	
B733	E	0.2380	0.2380	
B734	E		0.1797	
B735	E	0.2380	0.2380	
B736	D		0.2191	
B737	D		0.2191	
B738	D	0.2320	0.2320	
B739	D	0.2320	0.2320	
B744	B	1.1000	1.1000	[28]
B748	B	1.1000	1.1000	[28]
B752	C		0.2775	
B753	C		0.2775	
B762	C	0.2497	0.2497	
B763	C	0.3330	0.3330	
B764	C	0.2497	0.2497	
B772	B		0.6669	
B773	B		0.6669	
B77W	B	0.5734	0.5734	[29]
B788	B		0.6669	
B789	B		0.6669	
BCS1	D		0.2191	
BE20	F		0.3330	
CRJ1	E		0.1797	
CRJ2	E		0.1797	
CRJ7	E		0.1797	
CRJ9	E		0.1797	
CRJX	E		0.1797	
D228	F		0.3330	
D328	F		0.3330	
DH8A	E		0.1797	
DH8B	E		0.1797	
DH8C	E		0.1797	
DH8D	E		0.1797	

Model	CAT	Available Fuel (kg/s)	Taxi Fuel (kg/s)	Source
DHC6	F		0.3330	
E120	F		0.3330	
E135	E		0.1797	
E145	E	0.2200	0.2200	[30]
E170	E	0.1300	0.1300	[31]
E190	E	0.1500	0.1500	[31]
E195	E	0.1500	0.1500	[31]
E75L	E	0.1320	0.1320	[32]
F100	E		0.1797	
F50	E		0.1797	
F70	E		0.1797	
IL96	B		0.6669	
J328	E		0.1797	
JS31	F		0.3330	
JS32	F		0.3330	
JS41	F		0.3330	
L410	F	0.333	0.3330	[33]
MD80	D		0.2191	
MD82	D		0.2191	
MD83	D		0.2191	
RJ1H	E		0.1797	
RJ85	E		0.1797	
SB20	E		0.1797	
SF34	F		0.3330	
SU95	E		0.1797	
SW4	F		0.3330	
T134	E		0.1797	
T154	D		0.2191	
T204	C		0.2775	
YK40	E		0.1797	
YK42	E		0.1797	