UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Telecomunicació
i Aeroespacial de Castelldefels

TREBALL DE FI DE GRAU

TÍTOL DEL TFG: Cost index (CI) and take-off mass (TOM) estimation using machine learning algorithms

TITULACIÓ: Grau en Enginyeria d'Aeroports i Grau en Enginyeria d'Aeronavegació

AUTORS: Santiago Gil Vidal
Carles Olivares Guixé

DIRECTORS: Ramon Dalmau Codina
Xavier Prats i Menendez

DATA: June 15, 2017

Títol: Cost index (CI) and take-off mass (TOM) estimation using machine learning algorithms

Autors: Santiago Gil Vidal
Carles Olivares Guixé

Directors: Ramon Dalmau Codina
Xavier Prats i Menendez

Data: 15 de juny de 2017

Resum

El Cost Index (CI) i el Take-off Mass (TOM) són dos paràmetres molt importants per estudiar les preferències d'operació de les aerolínies. El coneixement d'aquests dos parametres permetria genera prediccions de trajectòries basades en terra de forma exacta. Avui en dia, desafortunadament, aquesta informació no és compartida per les aerolínies, ja que és informació confidencial perquè defineix les estratègies de mercat de la aerolínea.
L'objetiu d'aquest TFG és desenvolupar i evaluar un algoritme capaç estimar el CI i el TOM a partir de dades de la trajectòria de vol, que podrien ser recollides per una antena convencional (e.g. dades radar o ADS-B), utilitzant algoritmes de Machine Learning.
L'algoritme haurà de ser entrenat amb dades del PEP (Programa de Performance d'Airbus). Aquest s'enetrenarà amb milers de trajectories, variant la distància, TOM, CI i condicions atmosfèriques, per tal de constituir dades d'entrenament del Machine Learning. Un cop generat l'algoritme per garantir la seva robustesa serà testejat amb dades que contenen soroll alhora que s'evalurà la influència decadascuna de les variables de predicció. Finalment, l'algoritme amb noves trajectòries generades amb el PEP.
L'objectiu final del TFG serà comprovar i realitzar l'estudi amb dades de vol reals, per això s'obtindran dades radar provinents del DDR2, una plataforma de Eurocontrol. A partir d'alguns vols, es farà un estudi dels valors de CI i TOM utilitzats per diverses aerolinies amb l'algoritme de Machine Learning previament entrenat.
Com a conclusió s'ha pogut demostrar que el parametre més rellevant en la predicció del CI és el Número de Mach, ja que és la relació més directa amb la relació del cost temps-combustible. En canvi, pel cas del TOM, s'ha vist que està més relacionat amb les distàncies i nivells de vol (FL). A l'aplicar l'algoritme a casos de vols reals s'ha observat que les aerolínies de baix cost i les de bandera usen estratègies diferents de CI. Tot i així, una mateixa companyia acostuma a utilitzar el mateix valor de CI per a la majoria de rutes, desaprofitant l'oportunitat d'optimitzar els costos de la ruta i tots els avantatges que ofereix el CI.

Title : Cost index (CI) and take-off mass (TOM) estimation using machine learning algorithms

Authors: Santiago Gil Vidal
Carles Olivares Guixé

Advisors: Ramon Dalmau Codina
Xavier Prats i Menendez

Date: June 15, 2017

Overview

The Cost Index (CI) and Take-off Mass (TOM) are two parameters that are very important in order to study the preferences on airlines operation. In the same way, these two parameters would allow to predict ground-based trajectories accurately. Nowadays, unfortunately, this information is not shared by the airlines, because this information is confidential as they help to define market strategies of the airline.
The objective of this final degree project is to develop and evaluate an algorithm able to estimate CI and TOM from data of a flight trajectory, that could be collected by a conventional antenna (i.e. radar data or ADS-B), by using Machine Learning algorithms.
The algorithm should be trained with data from the PEP (Performance Program Airbus). The data will be shaped by thousands of trajectories generated with different ranges of distances, TOM, CI and atmospheric conditions in order to establish the input training data for Machine Learning. Once the algorithm has been generated, to ensure its robustness, it will be tested with data containing noise where the influence of the parameters in the prediction would be evaluated. Finally, it will be validated with new aircraft trajectories from PEP.
The ultimate goal of the final degree project is to check and perform the study with real flight data. To realize this, radar data will be obtained from the DDR2 platform of Eurocontrol. With some flights trajectories, we will study the values of CI and TOM used by several airlines with the Machine Learning algorithm previously trained.
In conclusion, it has been demonstrated that for CI the most relevant input variable is the Mach Number because it is the most visible evidence given to the time-fuel cost relation. On the other hand, TOM is more related to the distance of the flight and flight levels (FL). When the prediction algorithm is applied to real cases flights, we observed that low-cost airlines and flag carriers use different strategies of CI. Even so, a single airline usually use the same CI for most of their routes, wasting the opportunity to optimize the costs of the route and all the advantages offered by the CI.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Aircraft trajectory prediction has been always been a key issue for on-board and ground-based applications in Air Transport, but their importance will increase even more in a near future with application of SESAR (Single European Sky ATM Research) and NextGen concepts [2]. In this study, we apply Machine Learning to obtain some unknown parameters of flight trajectories.

In the last couple of years, the data-link between aircraft and ground-based systems has incredibly improved. The implementation of data-link permits ground-based systems to download on-board trajectory predictions. Even the progress is evident and notable, in some cases ground-based trajectory prediction is still necessary. An example could be found when Air Traffic Management / Air Traffic Control (ATM/ATC) have problems and require testing a large number of trajectories. In that situation, instead of downloading all aircraft trajectories, ground-based trajectory prediction will be a better solution.

Even ground-based system dispose of aircraft data (position, velocity, callsign...), there are many parameters still unknown because of the competitiveness between operators. Parameters like actual mass, thrust setting of the engines or the cost index of the route are not transferred to ground stations. Airlines prefer not to divulge strategic parameters which influences on the cost and time of their routes.

Some studies have already searched for these unknown parameters by applying Machine Learning methods. Two examples could be found in airspeed prediction during climb and mass estimation methods for ground-based aircraft climb prediction [3] [4]. This articles uses Machine Learning methods to predict flight phases or velocity, while in this project this methods are going to be used to obtain two flight parameters Cost Index (CI) and take-off mass (TOM).

The goal of this final degree project is to present an algorithm to estimated the CI and the TOM of an aircraft using machine learning and surveillance data. In aviation, the term CI is used to evaluate the impact of time and fuel cost on a route. It is a strategic parameter that has a relevant influence on aircraft trajectory and airline economy. Depending on its value, the three main phases of flight (climb, cruise and descent) would be modified. Also, TOM limits aircraft performances and its value could produce important changes on the flight trajectory.

This project has two steps. During the first step, the machine learning algorithm will be trained with known flight trajectories obtained from the Performance Engineer Program (PEP) of Airbus. During the second step, the CI and TOW of several trajectories will be predicted using the trained algorithm and its performance will be assessed by means of cross-validation. Figure 1 shows both steps of the project, and the stages that are done in each one of the steps.

Finally, the prediction algorithm will be verified with aircraft trajectories generated with new trajectories of PEP not used for the trainning. Thanks to the validation, it would be possible to assess the accuracy of the model and to do a final implementation of evaluate real affic trdata obtained from EUROCONTROL platform.

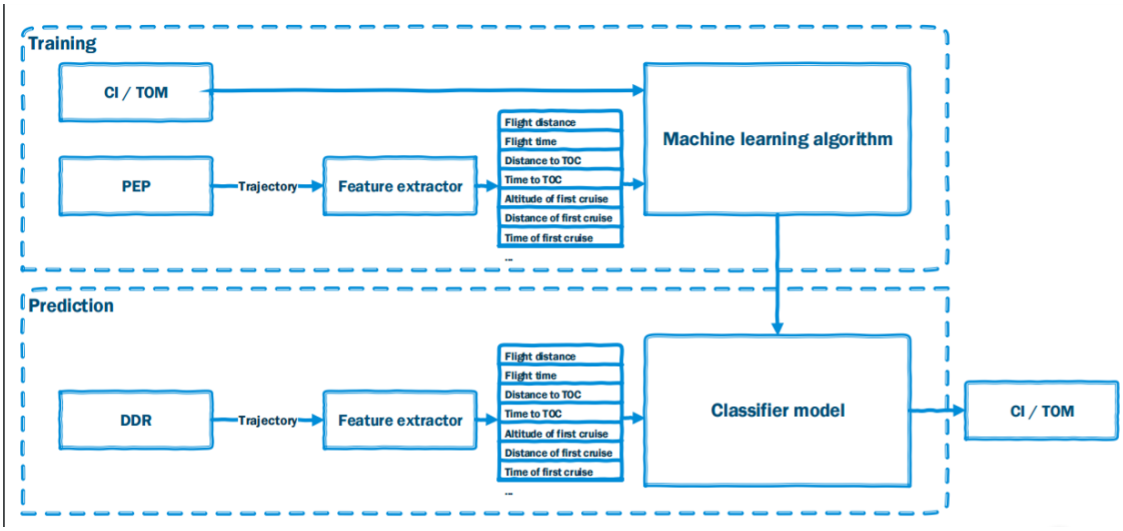


Figure 1: Machine Learning process

CHAPTER 1. BACKGROUND

Before to start the study, it is needed to introduce a brief explanation of the main concepts that will be used in the following chapters. Also, it is explained other relevant models related to the study.

First, we would respond to the following question; why is important to discompose a flight in different phases? Whenever is needed to analyze the information from a flight trajectory, it is useful to separate the whole flight in different phases. These phases have related parameters such as aircraft speed limitations or determined transition altitudes that give relevant information about other related flight variables which could be unknown. The goal of this final degree project is to estimate the CI and TOM from the characteristic altitude, duration and speed of the different phases of the flight, and to separate the flight information in different phases will be crucial for a better analysis of the routes.

On one hand, to obtain the CI value, sometimes it would be enough seeing the starting or the ending phase of a flight. It is strongly related to the climb gradient of the aircraft ascension at the climb phase or the descend gradient at the descent phase. But other times, it is needed an deeper analysis on cruise phase to search the points of changing altitudes or Mach cruise speed to get an accurate result.

On the other hand, characteristic weights delimit the values of possible TOMs. As the same way as CI, flight phases permitted to find relations between the aircraft trajectory and how the aircraft TOM is changing during the flight.

But, what happens if the day conditions vary from one flight to another? Flight parameters would remain constant or will depend on the weather of the day? In aeronautics, it was needed to model an atmosphere where all pressures and altitudes where established the same for all the aircraft. Consequently, it was created the ISA model which permits to unify meteorological conditions. When, climate conditions are far from the standard values stipulated, it must be applied an ISA deviation.

In order to have real flight simulations, the program selected to generate flight trajectories has been PEP. PEP is an Airbus program intended to many purposes related with flight planning, performance evaluation and noise computation, among others. PEP is composed by many modules, on of which can be used to generate the planned vertical profile (i.e. altitude and speed) of a flight given certain parameters. With PEP simulations, the algorithm will learn how each parameter affects the aircraft trajectory and, therefore, how the prediction variables change.

Finally, once the Machine Learning algorithm has been trained, the program will be tested with real flights. The aircraft trajectories will be obtained from DDR, an Internet platform from Eurocontrol, which collects all European Flights.

1.1. Flight Phases

A flight can be decomposed in to three phases. These phases are the following:

- **Climb:** the climb phase starts at the take-off and ends with the aircraft reaching the first cruise level at the top of climb (TOC). This phase, due to reach the best

optimization, is supposed to be a continuous climb (CCO). [5]

- **Cruise:** Is the main, and typically, the longest part of the flight. It is the phase where the airplane was designed for, where the minimum air resistance is obtained (Drag), and where is burned the highest quantity of fuel due to its distance and time required. The cruise is performed at constant altitudes (or flight levels(FL)). However, since the optimal altitude increases as the aircraft weight decreases, the aircraft may eventually climb to the following available flight level by performing a step climb.

- **Descent:** Is the last part of the flight, where the airplane starts release altitude and speed to reach the destination. The most efficient way to realize is a continuous descent approach (CDA). [5]

All these phases have some related velocities which mainly depend on the CI, the TOM and the weather conditions (i.e. temperature and winds). For a given combination of these parameters. Flight Management System (FMS) does not only compute the optimal vertical profile altitudes (Flight Levels), it also computes the optimal velocity profile, as can be shown in the Figure 1.1 . It is necessary to know which are the main velocities that are represented in a flight.

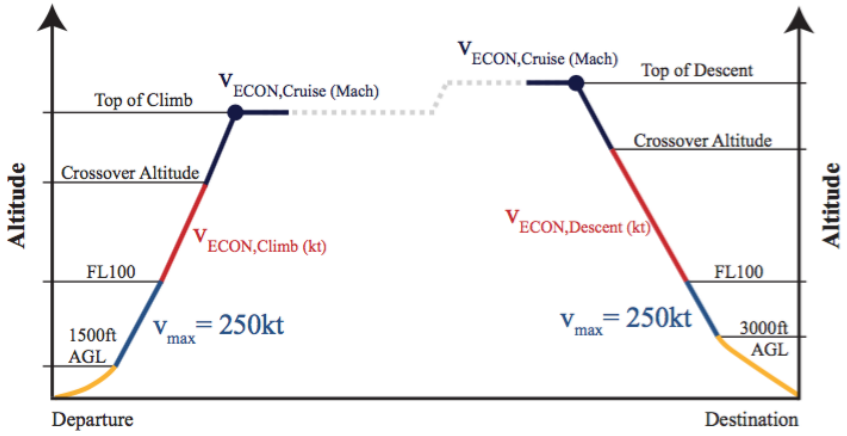


Figure 1.1: Velocities Profile

As the Figure 1.1 shows, the yellow lines represent the Take-Off and Landing phase. In this phase, the aircraft is increasing or decreasing its velocity and it will always be a curve line because the climb or decent gradient will not be constant. When the aircraft arrives to 250kt, it remains at constant velocity because there is a flight velocity limitation of a maximum of 250kt until reaching FL100. In the graph, this climb/descent phase is represented in a blue straight line. Once the flight overpass FL100, the FMS define a $V_{ECON,climb(kt)}$ that is the IAS velocity [kt], represented in red color in the figure, which allows the optimal climb. There is one moment, when the airplane reaches a crossover altitude[1], where the velocity starts to be given by Mach Number. The same happens in the descent phase where it is defined an $V_{ECON,Descent(kt)}$.

[1]The Crossosver Altitude is the altitude at which a specified CAS and Mach value represent the same TAS value.

Once the aircraft is flying above the crossover altitude, the aircraft starts working in Mach velocities. For each cruise altitude, it has a Mach velocity associated named as $V_{ECON,Cruise(Mach)}$, which is the optimal speed computed as function of the CI and the weight of the airplane.

1.2. Cost Index, factors and usage

Some airlines prioritises fuel saving while others prefer to arrive to the destiny in the minimum time possible. For that reason, it was needed to define a term, which relates both concepts. Cost Index (CI) is the ratio between the time-related cost of an airplane operation and the fuel cost.

$$CI = \frac{TimeCost \sim \frac{\$}{hr}}{FuelCost \sim \frac{\$}{kg}} \tag{1.1}$$

To do the study, it has been taken a range from 0 to 100 CI values for the case of Airbus A320. That is because PEP's Flight Management System (FMS) uses this values, but depending on the airplane FMS it can take values from different ranges, like 0-999 (for example Airbus A340). To a better understand of the CI relevance, if it is assumed to have a CI equal to 0, it will imply a route performed at minimum range speed and minimum trip fuel. And in the complete opposite, for maximum values of CI, FMS uses the minimum time speed without taking into account the fuel cost.
But from where did the CI relation come from? [6] First, a trip cost can be expressed as a sum of fix and variable costs:

$$C = C_{fuel}\Delta F + C_{time}\Delta T + CC_{fix} \tag{1.2}$$

Where: *C is overall cost [$];* C_{fuel} *is the unitary cost of fuel [$/kg];* ΔF *is the trip fuel[kg];* C_{time} *is the unitary cost of time [$/min];* ΔT *is the trip time [min];* CC_{fix} *is the cost independent from time[$].*

The way to optimize the overall trip cost is to minimize the variable cost, because are related with the flight performances. Dividing the expression of the total cost by the cost of the fuel, the cost function (τ) is expressed as:

$$J = \frac{C}{C_{fuel}} = \Delta F + \frac{C_{time}}{C_{fuel}}\Delta T \tag{1.3}$$

Where:

$$\frac{C_{time}}{C_{fuel}} = CI \tag{1.4}$$

From the term above, it is defined the CI as the relation between the time cost and the fuel cost. For a given route and the length of the trip is known, it can be computed the total cost of the unitary length as:

$$J(1Nm) = \frac{1}{SR} + \frac{CI}{G_s} = \frac{\Delta F}{\Delta SR} + \frac{CI}{G_s} = \frac{\Delta F + CI}{G_s} \tag{1.5}$$

Where:

$$SR(Specific \quad Range) = \frac{\Delta SR}{\Delta fuel} \quad [Nm/kg] \tag{1.6}$$

$$G_s = Ground \quad Speed = Wind + TAS \quad [kt] \tag{1.7}$$

If the cost function is integrated for all the range of the trip, the result is the total variable cost of the trip:

$$C_{Variable} = \int_0^{Range} \frac{\Delta F + CI}{G_s} dx \tag{1.8}$$

Cost Index is a parameter, which could determine important variations on aircraft Flight Plan. Actually, airlines use the FMS to compute the appropriate trajectory in function of the CI selected. However, airlines don't take full advantage of this calculation in all their routes.

CI depends on time trip and time is related to aircraft velocity. So, the main point is CI could introduce variations on aircraft speed. These variations are transcendental for aircraft performances and there are some studies which introduce the affection of CI on speed changes. For example speed change up to Mach 0.09 in cruise phase, which will correspond to approximately 10% of speed variation. The affectation on the climb and descent are also notable. It could reach values of 96 knots variation between different values of CI. Another example is how CI affects the descent is the distance between the optimum positions of the top of descent, which could differ up to 20 NM.

Airlines operating costs can be clearly reduced adjusting the CI value. This ratio between time and fuel cost permits operators to modify the CI for each route requirements.

On one hand, low values of CI should be used when fuel cost are high compared to other operating costs. On the other hand, for high values of CI time will be more relevant than fuel cost, so time will be prioritized searching for the minimum time flight. There two affecting factors for CI:

- **Time-related direct operating cost.**

  First factor does not include fuel cost and it is the numerator of CI ratio. In a flight, there are many items which have a dependency with the operating time such as flight crew wages or some aircraft maintenance. Some structural and mechanical items like the fuselage or engines should be revised periodically. When the number of hours flown is reached, there must be a maintenance associated. Maintenance costs can be accounted for on airplanes by hours, by the calendar, or by cycles. Those items which depend on operating hours causes high direct cost on time flight.

  The cases where time-related cost is high, airlines should choose larger CI to minimize time flying. In the other way round, when costs are fixed time flight is not so important because there are no extra cost associated, so airlines would search for low CI.

- **Direct fuel cost.**

  Fuel cost is the denominator of CI ratio. As fuel cost is not fixed, it is another parameter to take into account. One important issue to be known is fuel tank does not need to be full each time there is a flight. Depending on the distance range and the following aircraft routes, a necessary fuel calculation could be made. Aircraft must have enough fuel to perform their nominal route, plus the trip reserves and alternate reserves.

  As less weight carries the airplane, less consumption fuel. So, it would be normal to think that in each route the plane should be filled only with the required fuel to perform that route. Then, once landed, the plane could fill the fuel tanks again and proceed to the next destination. This would be true if we do not care about fuel price difference between countries. Depending on the country, there is an extreme difference between fuel cost. For example, there are situations where airlines could prefer to fill the fuel tank to the maximum even the fuel consumption during the trip would be higher. This is because the fuel price at the origin is exceedingly lower than at the destination.

Also, CI values affect on flight phases. For aircraft climbing, the steepest slopes are for lower CI because the aircraft wants to go up as soon as possible to reduce fuel consumption, seen in Figure 1.2. For larger CI, the slope decreases to maximize time climbing.

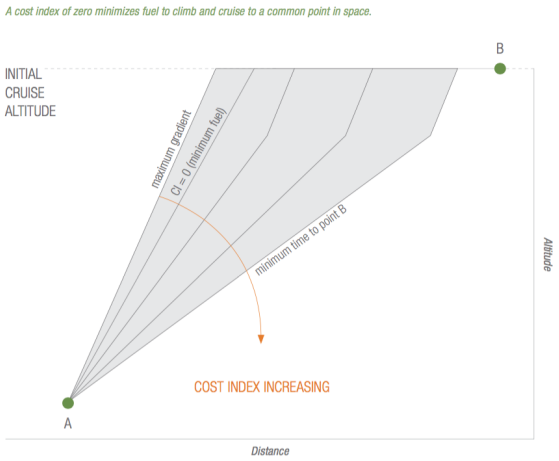


Figure 1.2: CI effects on climb phase [1]

For aircraft descents, the earliest Top Of Descent (TOD) is found for lower CI, as seen in Figure 1.3. This point permits a continuous descent with minimum thrust and reducing the fuel consumption. For minimizing time and obtaining larger CI, consists on being as much time as possible on cruise where velocity is optimized.

Figure 1.3: CI effects on descent phase [1]

1.3. Characteristics Weights

The airplane manufacturer define some weights that are relevant for the performances of the flight, those weights have to be certified for the correspondent aeronautical authority to ensure the airworthiness of the airplane, the most relevant ones are:

- **MTOW:** Maximum Take-Off Weight. Maximum permissible weight when starting take-off for standard atmosphere conditions. Limited by aircraft structural and aerodynamic conditions.
- **MLW:** Maximum Landing Weight. Maximum permissible landing weight. Limited by structural strength of the aircraft when impacting against the ground and aerodynamic conditions.
- **MZFW:** Maximum Zero Fuel Weight. Maximum permissible weight before fuel loading.
- **OEW:** Operation Empty Weight. Weight of the aircraft needed to operate. Weight of the structure, engines and equipment that can be considered as an integral part of the aircraft configuration. Includes necessary personnel, equipment and supplies except fuel and payload.
- **MFW:** Maximum Fuel Weight. Weight of the maximum amount of fuel the aircraft can load. This limit is the volumetric capacity of the aircraft's deposits.
- **MPL**: Maximum Payload: It is the maximum burden that brings benefits to the company. Includes passengers with their luggage, mail, parcel and cargo. The MPL limit is set by the structural strength of the aircraft:

$$MPL = MZFW - OEW \tag{1.9}$$

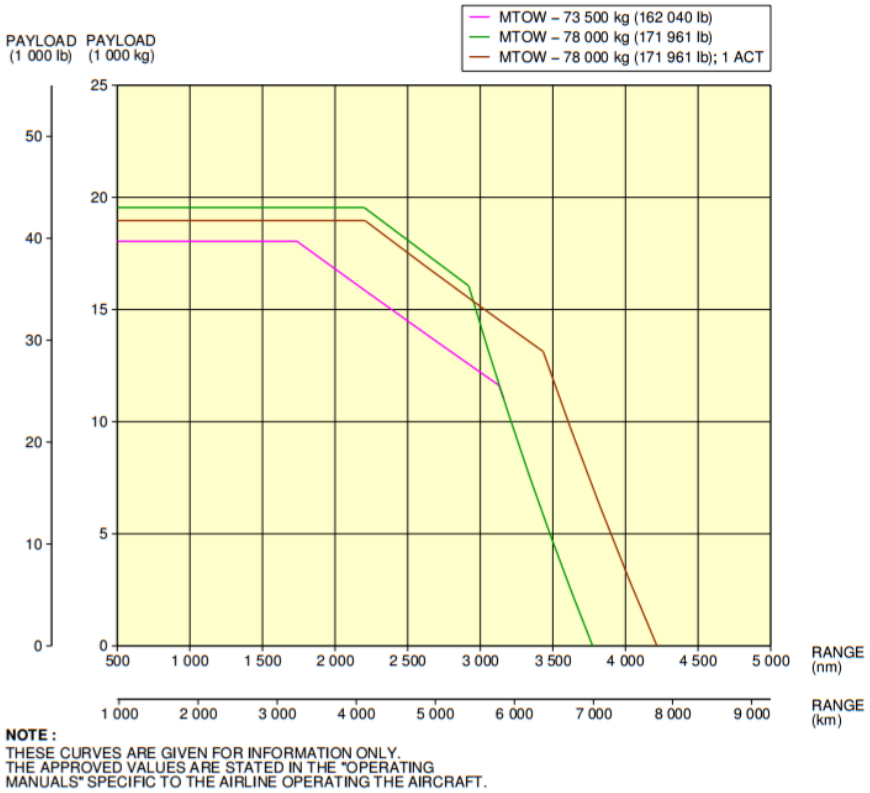


Figure 1.4: Payload vs. range on A320-200

Figure 1.4 shows the payload-range diagram of the A320-200. As it can be seen, payload limits range when the aircraft is loaded with MTOW. If aircraft payload is reduced maintaining MTOW, fuel quantity increases which implies an increase of range. Finally, the maximum range is acquired when there is no payload on board and fuel quantity on board is equal to MFW.

In Appendix A there is defined the characteristics weights for our case study.

1.3.1. Weight limitations

Weight would be an important parameter in the study, so it must be well explained. Characteristic weights are imposed by the aircraft manufacturer, which will always be delimited by the minimum weight possible (OEW) and the maximum (MTOW). As the range is stipulated by the manufacturer, the range values will go from a minimum range value depending on the aircraft model to its maximum. In this case, we will use data from Airbus characteristic weights as the aircraft manufacturer. But even ranges are defined, there are many other parameters which could change the possible weight values.

When generating the files to train the Machine Learning, there could be some TOM values that are not valid. Sometimes, short distance delimits the maximum TOM to not exceed other characteristic weights such as MLW or MZFW. Other times, large distance could exceed the MFW or need a negative Payload to complete a long trip. All the case studies which does not fit a realistic TOM value, must be remove and not used for the training.

When talking about operating altitudes, although distance range has the greatest affectation, weight also limits the maximum altitude. As more TOM, aircraft will need more lift to be sustained and can not reach the maximum altitude possible. Once TOM decreases, it permits aircraft to climb to its maximum stipulated altitude.

1.4. ISA Model

To a better understanding of some phases of the study, it will be introduced a brief concept of ISA model. Civil Aviation Organization defined a hypothetical model called International Standard Atmosphere (ISA)[7] which corresponds to an ideal atmosphere based on thermodynamic equations which relates pressure to altitudes, as can be seen in Figure 1.5. It uses a standard reference for pressure, density, viscosity, and temperature at different altitudes throughout the atmosphere. Therefore, ISA is a model which express the temperature, pressure and density as a function of the altitude. Standard sea level pressure and temperature are 1.013,25 mb (P_0) and 15°C (T_0).

For the cases, that the meteorological conditions are not equal to the ISA conditions, it is said that we are at ISA± X. For example, at sea level (SL), if actual temperature is 20°C, 5°C above the standard, we are in ISA+5.

The ISA model is expressed by the following equations:

$$T = T_0 - 1.98\frac{h(ft)}{1000}^{2} \tag{1.10}$$

$$P = P_0\left(1 - 0.0065\frac{h(m)}{T_0(^{o}K)}\right)^{5.2561} \tag{1.11}$$

$$\rho = \frac{P}{RT} \tag{1.12}$$

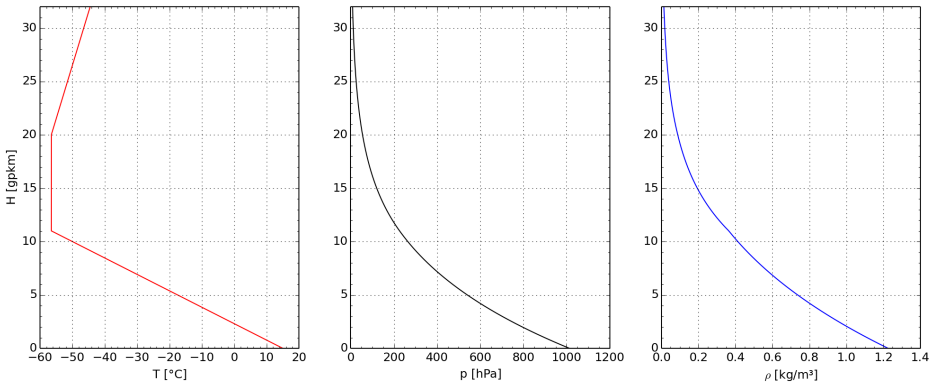


Figure 1.5: International Standard Atmosphere vs. height

Where: *T is the temperature [K];* *P is the pressure [Pa];* ρ *is the air density [kg/m^3];* *R is the real gas constant for air [287,04m^2/Ksec^2].*

[2]Equation applicable only up to Tropopause (h=11.000 ft)

1.5. PEP: Performance Engineer's Program

Flight trajectories are taken from a flight-planning program called PEP (Performance Engineer's Program). Airbus has a module where it can be computed trajectories for different aircraft types and scenarios in order to make a flight planning. The most efficient route in terms of cost or time, depending on the Cost Index fixed by the company, will determine its Flight Plan. Even the program has many modules, the study has been performed with a tool called FLIP.

FLIP permits to adjust flight parameters and simulate trajectories in a real way. FLIP has only Airbus aircraft types, so the study is centred on Airbus models commonly used by airlines. Aircraft specifications are included in the tool, so performances and weights are taken in account during the simulations. Among other parameters, FLIP considers aircraft's limitations on weights, velocities and altitudes. Also, the user can introduce variations respect ISA conditions and wind influence. All this inputs can help the user to recreate the most realistic scenario possible.

An important matter is to use FMS flight planning simulations. In FLIP, for a standard flight cannot be add a cost index value. Without cost index fixed, velocities will be modified as the program optimizes fuel consumption. FMS permits to add a cost index value for the flight and it has in consideration all the speed limitations under ATC regulations and MMO velocity.

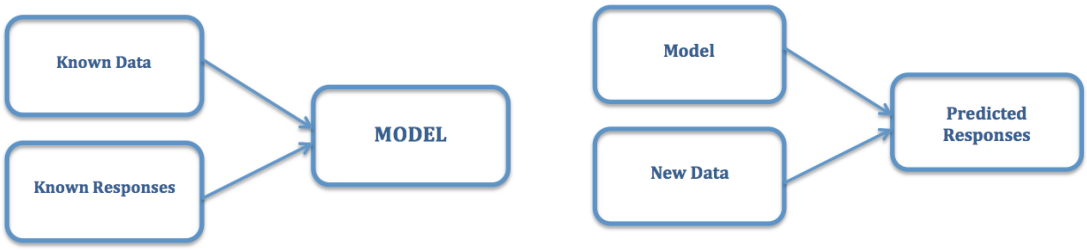
In this case study, we have used the Airbus aircraft models that are available in PEP performance program. All the properties of the airplanes used are detailed in Appendix A where data have been obtained from official Airbus web page.

1.6. Machine Learning

Sometimes, when you are generating a model, there are parameters which are unknown. Machine Learning techniques and algorithms use computational methods to learn information directly from data experience. Machine Learning teaches computers to learn from past data and it is able to predict future values for the unknown parameters. Algorithms adaptively improve their performance as the number of past samples introduced increases.

There are two types of techniques of Machine Learning: supervised learning and unsupervised learning. In our case study, we used the first one, which trains a model on known input and output data to search for a future predicted outputs.

First, it is needed some known data about the predictor and response variables are needed to train the model. Figure 1.6(a) shows how is added known data with the responses to generate the model. This model gains confidence and is increasingly able to predict more accurate responses by adding new known data. Once the model is trained, Figure 1.6(b) shows how new data could be added to the model, which will predict the new responses for the unknown parameters.



(a) Model generation (b) Predicted Responses

Figure 1.6: Machine Learning process

Our parameters that need to be predicted are Take-Off Mass (TOM) and Cost Index (CI). This is because there is no information about their values and, except for the aircraft company, they are commonly two unknown parameters of the trajectory.

To compute the prediction variables, we used Machine Leaning from Matlab Toolbox [8]. Matlab propose different methods but there is not a unique solution. Finding the right algorithm is partly just trial and error. Even though, depending on the size and type of data introduced, there are algorithms which have a better fit.

1.7. Demand Data Repository (DDR)

This is a platform crated by Eurocontrol [9] with the Objective to provide the most accurate picture of pan-European air traffic demand, past and future.

The DDR project was divided in two phases:

- DDR1: It is currently being phase out. This produce future traffic samples, mainly using historical traffic samples adjusted with STATFOR forecast data.
- DDR2: covers DDR1 functionalities and also collects early available flight intentions from airlines (SSIM/INNOVATA data) and from coordinated airports through the European Union Airport Coordinators Association (EUACA).

In the DDR2 platform , there is a historical report of all European flight collected. In the Figure 1.7, there is an example of the DDR2 platform to show how is presented. All flight data is presented in a table, where is classified according to corresponding day of the month. Each row presents a different file format to be downloaded and it is also added the number of flights per day and its ranking position of the month.

As it has been seen in the platform, flight data is presented in a table where users can download the flights in different file formats. But it is important to know the difference between the two types of traffic:

- .m1: This document contains all the flight plans that was presented in a certain period, chosen by the user, and those flight plans presented to the FMU (Flight Management Unit) an were approved. It uses the last saved flight plan after all the modifications with IFPS messages.

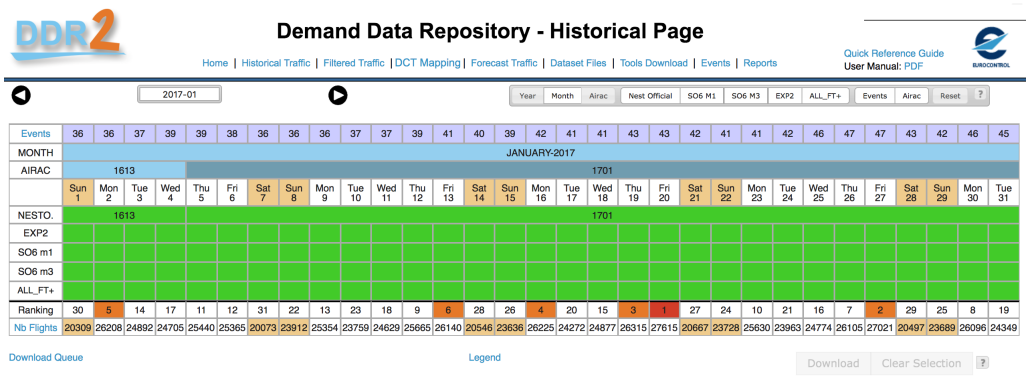


Figure 1.7: DDR2 platform

- .m3: This document contain all the flights realized during a certain period. It uses the actual 4D trajectory recalculated by aligning it to the existing route points.

CHAPTER 2. AIRCRAFT TRAJECTORY GENERATION WITH PEP

As it has been explained before in section 1.5. PEP is composed by several tools which can be used for many purposes (flight planning, performance data generation, noise calculation, etc). For the case study is used FLIP mode with FMS, which has implemented all speed regulations and tries to adjust aircraft speed to its optimal value.

First of all, it is essential to decide which aircraft will be studied. From the list of available aircraft, will be studied the Airbus A320. A320 is the aircraft model most typically used by European airlines. For this aircraft model thousands of trajectories with different values on **TOM, CI, Range, TOCs, Wind,** ΔISA will be computed. These data will be used as known data to let the machine learning algorithm learn from these trajectories.

As the number of trajectories to be introduced in PEP is over 100,000 samples per aircraft model, the program cannot be manually executed. In order to generate the PEP input files, it has been used an script with loops changing the values of the different flight parameters explained before.

2.1. Script: Input files generation

The script permits to generate the PEP input files where each file will correspond to a unique trajectory. There are some conditions that must be verified before a trajectory is added to the study. Performance parameters could oscillate due to the aircraft model or flight conditions. The following chapters present some limitations used to filter the possible trajectories for the study.

2.1.1. Relation between variables

Even it has been defined an adequate range of values for the variables of each type of aircraft, some results will not be valid. This is because there is a dependence between some variables, which could generate non-possible trajectories. An example would be the limitation on Range and TOC due to the aircraft TOM. Both payload and fuel on board are factors which affects directly to route range. When aircraft carries MPL, the maximum possible range of the aircraft is reduced. A similar thing occurs with TOC, which could not reach its maximum altitude on cruise because of the heavy weight.

Other parameters which are interrelated could be ΔISA and wind with fuel consumption. Once it is known the use of the ISA model, section 1.4. explained before, it is important to notice that atmospheric pressure and temperature decreases with height at a standard lapse rate. But what happens when the real pressure and temperature of a concrete day does not correspond to the theoretical ISA values? Here is the point where ISA deviation appears (ΔISA), to adjust the pressure/temperature of the day to the corresponding real altitude of the aircraft. Aircraft performances will vary depending on the height. An ΔISA could affect to fuel consumption, and so to CI calculation. Also, the wind is a factor to be taken into account, because it will modify the ground speed of our aircraft and, with it, the total time and cost of the trip.

The range values of the variables for the selected aircraft have been added in Appendix B.

2.1.2. Affectations on range variables

Apart from the relation between variables, there are some factors like the aircraft model or meteorological conditions that will also affect loop ranges. Although some variable will be fixed, others will not have constant range values.

Wind and ΔISA are meteorological affectations on aircraft. In this study, the aircraft model flies in flight levels between 20,000 ft and 43,000 ft. As altitudes do not overpass the tropopause, air conditions could be considered lineal, the temperature drops as -2^{o}C/1.000ft. In this point, we set the ranges in usual values for wind speed (between -80 kt and 80 kt) and for ΔISA (between -20 and 20).

Aircraft models will affect on a different way. Characteristic weights and aircraft range are imposed by the aircraft manufacturer. For characteristic weights, range values will always move between OEW and MTOW. Also, the the maximum operating altitude, also known as aircraft ceiling, is imposed by structural and motor limitations. It will set the possible TOCs of the aircraft and define the ranges of possible altitudes for each aircraft.

To set CI values, it has been defined a fixed range for A320 model. Taking values from Airbus model, the aircraft has been studied for CI values from 0 to 100. In the cases of wide-body aircraft from Airbus, could be reached CI values of 999.

2.2. Input format file for PEP

The Python code generates two input files in different formats. The first file is in .dat format and contains all the parameters of the aircraft for the flight. The other is in .pep format and it is the file necessary to execute the FLIP program. This last document contains the routes of the input and output files generated before and after the PEP execution.

To execute the PEP program with more than one file at once, it has been used the Batch Manager. Batch Manager is an internal tool from PEP which permits to load many input documents to be executed on FLIP format.

An example of both format files are shown in Appendix C.

2.3. Reading the output file from PEP

Once the Batch Manager executed the input files of the aircraft, output files are generated. Output files contain all the information introduced in the input file an additional flight planning in a table format at the end of the document. The flight planning contains all information of the aircraft trajectory, but for the study we have focused on the variables explained before.

For this case study is necessary to define the variable required for machine learning. For that is necessary to analyze the vertical profile obtained from the PEP (Appendix C).

The FMS with the inputs values of range, take-off mass (TOM) and the Cost Index computes the optimal vertical profile. This vertical profile can define up to a maximum of four cruises altitudes, as can be seen in Figure 2.1, that is because the airplane loses weight as it burns fuel.

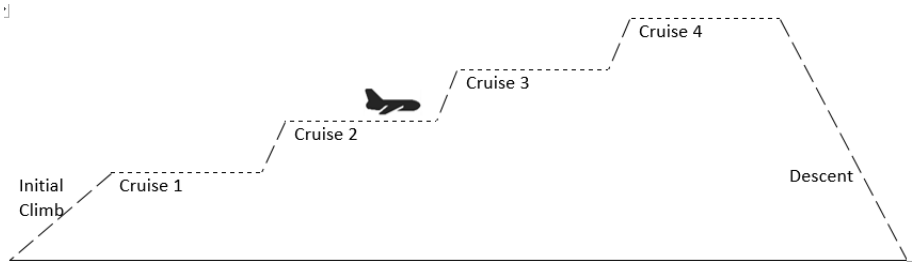


Figure 2.1: Flight Phases

For machine learning training, it is necessary to define variables that can be obtained from any trajectory to finally obtain the Cost Index and TOM. The variables defined that can be obtained from PEP output are the following:

- **Range:** The overall distance of all the flight. (dist_total [Nm]).
- **Trip time:** The overall time of the flight. (time_total [min]).
- **Climb:** Phase from take-off to first cruise from here we obtain the distance and time (dist_toc [Nm], time_toc [min]).
- **Cruises:** Main phase of the flight can be defined a maximum of four, in each cruise we obtain a total of four variables: the time, the distance, the flight level, Mach velocity. (dist_crN [Nm], time_crN [min], h_crN [FL], M_crN [Mach], where N is the number of cruise).
- **Descent:** the last flight phase, is from last cruise to landing here we obtain the distance and the time (dist_tod [Nm], time_tod [min]).
- **TOM:** the take-off mass used for the trip.
- **CI:** the cost index used for the trip.

The total number of variables obtained from the OUTPUT file is 24[1], and they are organized in a table, the table format is shown in Table 2.1, that is the best way to introduce these variables in the machine learning algorithm.

[1] In following chapters will be studied the possibility to add Wind and ISA conditions due to an improvement of the accuracy, making 26 the number of variables.

	dist_toc	*time_toc*	*dist_cr1*	*h_cr1*	*time_cr1*	*M_cr1*	...	*dist_crN*	*h_crN*	*time_crN*	*M_crN*	*dist_tod*	*time_tod*	*dist_total*	*time_total*	*TOM*	*CI*
Flight 1																	
Flight 2																	
...																	
Flight N																	

Table 2.1: Example of the table format

CHAPTER 3. MACHINE LEARNING DEVELOPMENT

This chapter presents the machine learning development in this project. The aim of this chapter is to train the algorithm to be able to compute the CI and the TOM with a data input of the flight trajectory.

Once the output files from PEP have been read, as explained in section 2.3., we pick up the significant information. The imput of machine learning in Matlab must be in a table format so we generate a table with the 24 variables that are necessary to start the training.

3.1. Matlab Machine Learning types

Machine learning of Matlab provides a toolbox that is able to learn from the experience using computational methods [10]. Machine learning uses two types of techniques; one that works with input and output data and the other that find hidden patterns in the input data[8] [11].

In the following sub-chapters are explained both types, the process of the selection of the algorithm can be seen in Figure 3.1.

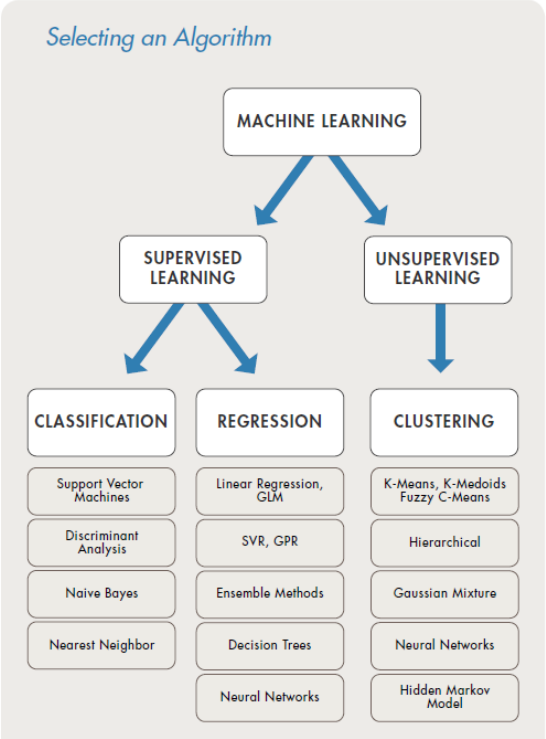


Figure 3.1: Matlab machine learning types

The process is a trial and error, and the algorithms are characterized by the following properties:

- **Speed of training:** Time required to generate the algorithm to perform the predictions.
- **Memory usage:** Part of the physical memory of he CPU destined for the process.
- **Predictive accuracy on new data:** Average hits obtained in predictions.

- **Transparency or interpretability:** How easily you can understand the reasons an algorithm makes its predictions.

3.1.1. Unsupervised Learning

Unsupervised learning is useful when you want to explore your data but don't have a specific goal yet or are not sure what information the data contains. It's also a good way to reduce the dimensions of your data.

Unsupervised learning finds hidden patterns or intrinsic structures in data. The most typical used is Clustering, which that basically consists on group data in clusters.

Example: *Most useful application is clustering, like gene sequence analysis, market research, and object recognition.*

This technique of machine learning is not useful to our case because all the data of our flight plans are previously filtered and we do not want to obtain groups of data. What is required in this case is to obtain the values of two variables (CI, TOM) based on some trajectories parameters.

3.1.2. Supervised Learning

The aim of supervised machine learning is to build a model that makes predictions based on evidence in the presence of uncertainty. A supervised learning algorithm takes a known set of input data and known responses to the data (output) and trains a model to generate reasonable predictions for the response to new data.

The main types for supervised learning are decision trees, classification and regression techniques.

3.1.2.1. Classification techniques

Classification techniques predict discrete responses. Some methods for classification are the Nearest Neighbor Classifiers, where in Matlab toolbox can be found some examples:

- **Support Vector Machines:** (SVM) Classifies data by finding the linear decision boundary (hyperplane) that separates all data points of one class from those of the other class. The best case to use this method is when data is linearly separable between two classes [12].
- **Discriminant Analysis:** Discriminant analysis classifies data by finding linear combinations of features. Discriminant analysis assumes that different classes generate data based on Gaussian distributions.
- **Neural Networks:** Inspired by the human brain,the network is trained by iteratively modifying the strengths of the connections so that given inputs map to the correct response. Permits to model highly non-linear systems, and also let the model be updated constantly with new data.
- **Naïve Bayes:** A Naïve Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. It classifies new data based on the highest probability of its belonging to a particular class. Their application is mainly used for small datasets with large number of parameters.

- **Nearest Neighbor:** (kNN) categorizes objects based on the classes of their nearest neighbors in the dataset. kNN predictions assume that objects near each other are similar. Distance metrics, such as Euclidean, city block, cosine, and Chebychev, are used to find the nearest neighbor.

- **Decision trees:** A decision tree lets you predict responses to data by following the decisions in the tree from the root (beginning) down to a leaf node. A tree consists of branching conditions where the value of a predictor is compared to a trained weight. The number of branches and the values of weights are determined in the training process. Additional modification may be used to simplify the model. It is used when an algorithm is easy to interpret and fast to fit, without a high predictive accuracy. There are different types of decision trees depending on the complexity of the data set in the model:complex, medium or simple trees [13].

There are other classification techniques which uses ensemble classifiers to predict the response.The ensemble method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model. The two examples of ensemble classifiers most used are:

- **Boosted Decision trees:** Boosted decisions involves creating strong learners by a set of weak learners by adjusting the weight of each weak learner to focus on misclassified examples.

- **Bagged Decision trees:** A bagged decision tree consists of trees that are trained independently on data that is bootstrapped from the input data. In this way bagged decision trees decrease the variance helping to improve the stability and the accuracy. It is a good solution to use bagging when input predictors are discrete or behave non-linearly.

3.1.2.2. Regression techniques

This techniques predict continuous responses. The Matlab toolbox presents the following ones:

- **Gaussian Process Regression:** (GPR) models are nonparametric models that are used for predicting the value of a continuous response variable. It can be applied in cases where there must be an interpolation of spacial data in presence of uncertainty.

- **Support Vector Machines:** (SVM) in regression algorithms find a model that deviates from the measured data by a value no greater than a small amount. It is able to predict a continuous response with a small error of sensitivity. It is very useful for high-dimensional data, where many number of variables needed to be predicted.

- **Linear Regression, GLM:** A Generalized Linear Model is a special case of nonlinear models that uses linear methods. It involves fitting a linear combination of the inputs to a nonlinear function (the link function) of the outputs.

- **Regression Trees:**Decision trees for regression are similar to decision trees for classification, but they are modified to be able to predict continuous responses. They are used when predictors are discrete or behave non-linearly.

3.1.2.3. Practical cases of Machine Learning techniques

For the use of classification techniques, the data set must follow some requirements to get an accurate prediction model. To obtain the best accuracy as possible, the model must have enough training examples, a good performance on the training set and should not be too complex. The figures below show **bad examples** of training data for classifier models:

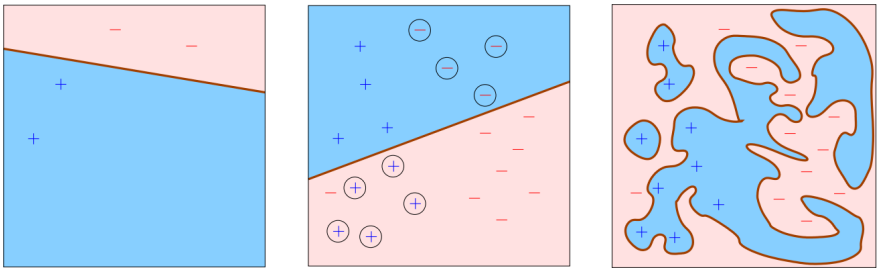


Figure 3.2: Bad examples of classifiers

In the order of images appearance, the one on the left present insufficient data to complete an accurate classifier model. Many zone on the field would be unknown and the model would do blind predictions. The image of the center present too much dispersion to adjust the model to accurate predictions. The training error would be too high, so accuracy would dramatically drop off. Finally, the image on the right, shows a very complex model would be very difficult to predict.

A classifier with high accuracy on predictions should have sufficient data, low training error and the data must be prepared as simple as possible. Here, there is a **good example** of classifier:

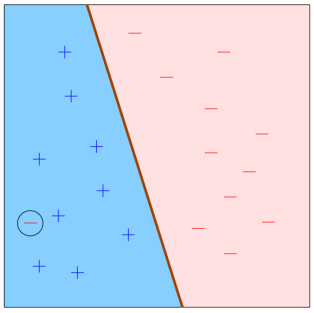
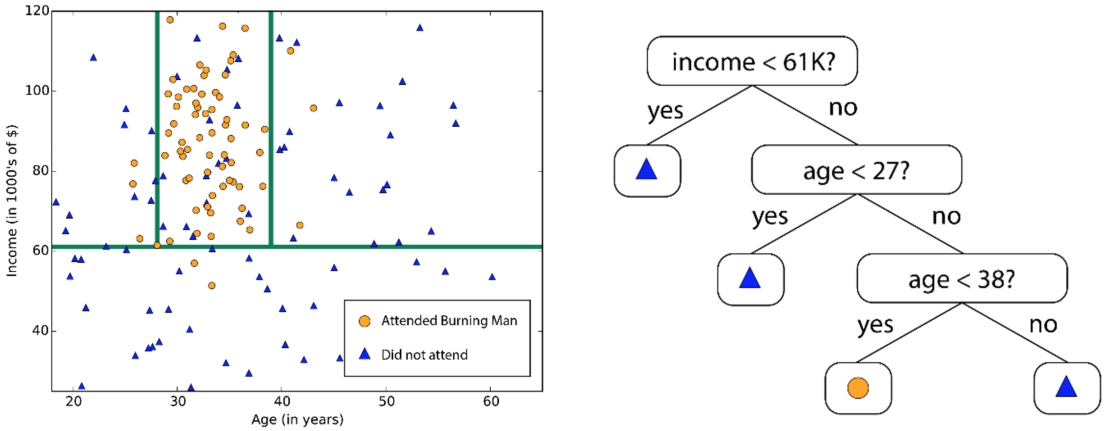


Figure 3.3: Good example of classifier

The use of classification techniques are diverse. Some examples are text categorization for spam filtering, fraud detection, optical character recognition, face detection, spoken language processing and understanding, market segmentation for customers promotions, between others. To a better understanding of how classifiers works, some practical examples are presented below. The algorithms chosen are the most common used and will be useful for our future study of CI and TOM prediction.

First, a practical example of a Decision Tree is exposed [14]. The study consists on the attendance to the USA festival Burning Man. The data set presents people's income in function of their age. In the Figure 3.4(a), data can be clearly separable in different groups between the green straight lines. In the right, Figure 3.4(b), the decision tree conditions permit a prediction response for each value of the whole model.



(a) Plot data Decision Tree

(b) Decision Tree

Figure 3.4: Decision Tree practical example

Another practical example for SVM classifier can be seen in Figure 3.5. The study wants to distinguish between sheeps and goats in function of mean daily temperature and steps per day they realize. Data can be cut in two main zones. Zones are separated for a border line with certain margin of non-confusion, where no sample could be inside.

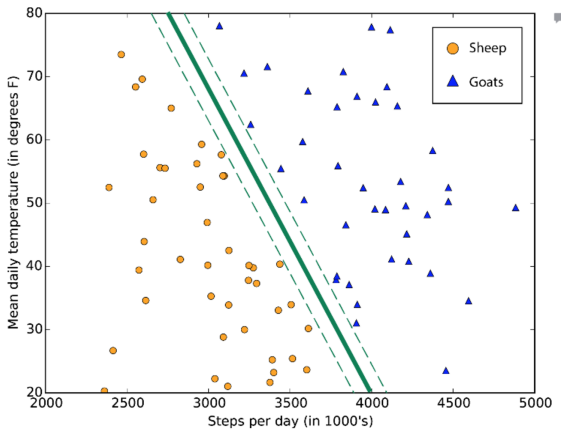


Figure 3.5: SVM practical example

The last example is Bagged Decision Tree. Figure 3.6 is not a practical case, but will permit to see how the algorithm works. It builds multiple such decision tree and amalgamates them together to get a more accurate and stable prediction. Once known the feature (f) from the sample, the algorithm predicts the solution for each tree.

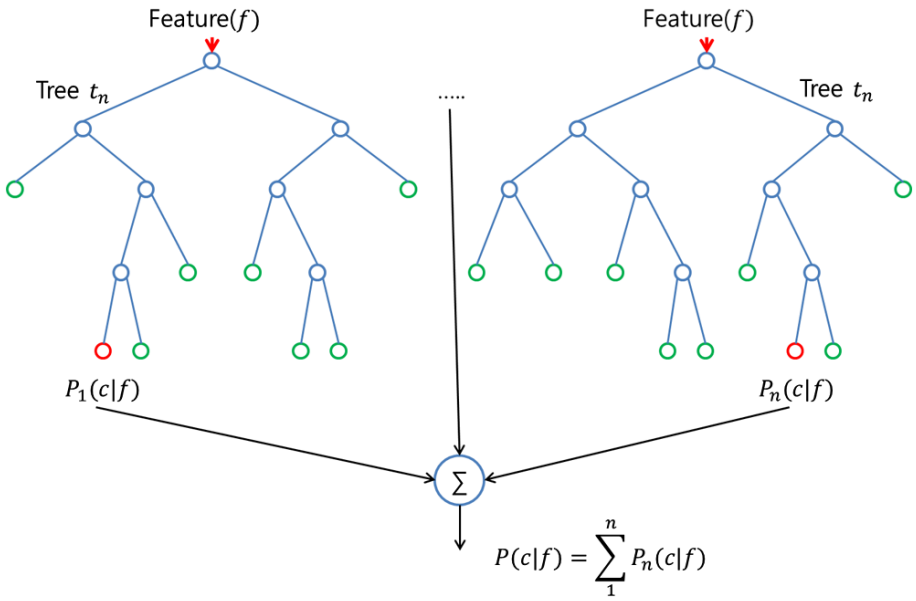


Figure 3.6: Bagged Decision Tree example

3.2. Training of the algorithm

To develop a Machine Learning some considerations have to be taken into account. The main steps and advises to ensure the best prediction are the following ones:

1. **Getting data:** The concept of 'more is more' is relevant in this part, as bigger is the data used for the training more accurate will result the algorithm. The training data must follow the same format of the tested one. Finally, the data must be truthful and accurate.

2. **Choosing features:** Using your own knowledge to know what features would be the most helpful is the first step. Redundant features do not affect the algorithm and some times can be helpful, because most of the algorithms do not require of independent features. But in some cases can generate overfitting and this redundant features can affect negatively to the prediction. If there are too many features, there methods of feature selection that would simplify it, as can see below.

3. **Choosing an algorithm:** Before to start, the learning paradigm has to be identified, that is to say, determine if it is a regression or classification problem, if the function is to predict a range or a value, and some others. In general, no learning algorithm dominates all others on all, it depends on the simplicity of the problem, that is the reason why many algorithms have to be tested, and it is a trial and error procedure. Here parallel compilation is useful such a way to improve the computing time.

4. **Selecting Parameters:** Imply a sensibility study of the parameters selected to make the prediction. The best method to do it is a trial and test by setting different parameters combinations.

5. **Testing Performance:** Train part of the data, and test on rest, that will allow to predict the accuracy of the algorithm, that method is cross-validation. Repeat the tests with split data in a natural mode (i.e. date split, depending on the date when data was obtained).

6. **Running experiments:** Once the algorithm is chosen and developed it is time to try to make predictions. To do it in an efficient way is to automate it, generate a script that allows

the reading of the data and to insert it to the algorithm, and finally returns the results in a specified format.

3.2.1. Training for CI and TOM

For our case study, once the table of the data is formed it is time to introduce it in the machine learning toolbox of Matlab to train the prediction model.

The data introduced will be the case study for A320. And for this data, there will be two Machine Learning algorithms because with one computation only can be subtracted one variable of prediction. For this reason, one algorithm will be for obtaining CI and the other one for TOM.

When the table is introduced in the toolbox, automatically detects the variables, its range, and offer the possibility to classify the variables in predictor, response or not relevant. Non relevant will be useful for those airplanes that not perform 4 cruises, with this method we can delete data of this for cruise if the ranges of the data are zero. Also, a study of the variables sensibility will be perform due to improve the algorithm accuracy, also will be considered the possibility to add atmosphere conditions (wind and ISA).

Also this toolbox provides us a confusion matrix, that shows the accuracy as function of the value of the predicted variables.

3.2.2. Avoiding training errors

Some algorithms could vary their accuracy due to their complexity. Data set must follow some requirements to get an accurate prediction model and avoid training errors. An example could be seen with decision trees. The tree must be big enough to fit training data, so the true patterns are fully captured. But, if the tree is too big, the data may overfit and could generate false patterns or capture noise due to have many branches in the tree. A graphical explication can be seen in Figure 3.7.

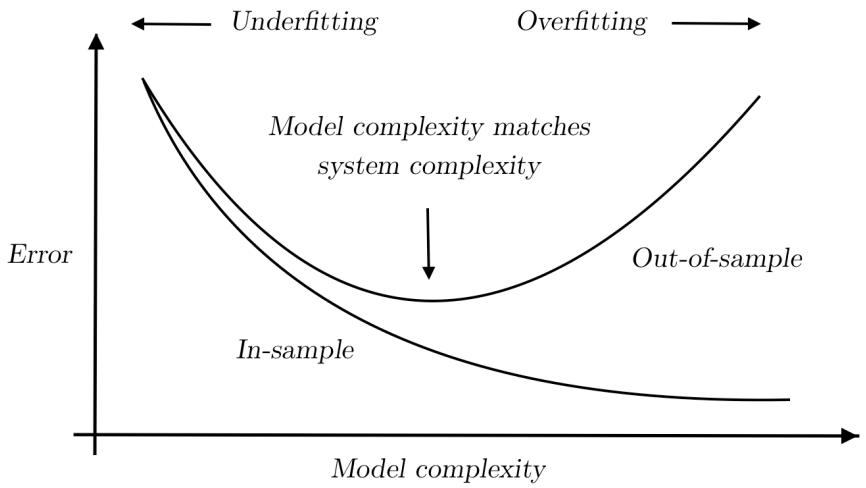


Figure 3.7: Underfitting and overfitting error

For example, when we try to approximate a data set of points to a polynomial function, can be seen the importance of the overfitting. In Figure 3.8, the data set points are fit with some polynomials of a different degree. While, in the first figure the points are not well approximated, as the degree increases, polynomial fits better the model. But, when the degree is increased too much, the complexity of the polynomial function is too high and may overfit, like it happens in the last figure.

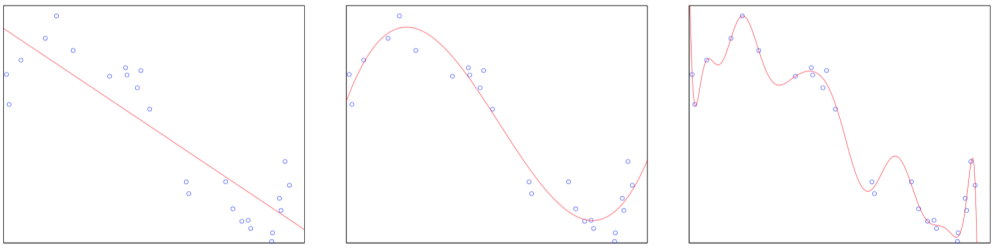


Figure 3.8: Polynomial fitting

Another problem which may occur with decision trees is the oversize of the tree. The error introduced by changing the number of branches in a tree, may vary depending on the data set. The best solution is to stop increasing the number of nodes in a tree when the training data accuracy is the optimal value, that correspond to the minimum error. But there is a significant problem to find this value because it can not be found the best tree size from the training error. So, the way of finding the stop training point is searching when the error begins to increase.

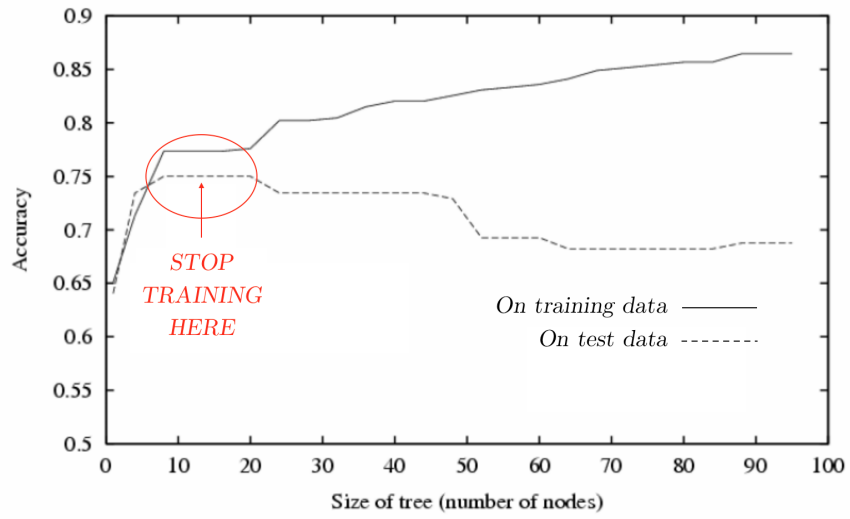


Figure 3.9: Decision trees size

The best way to find an optimal solution is to balance between simplicity and fit to data. Figure 3.9 shows the balance of the accuracy in function of the size of the tree. Depending on the algorithm used, parameters must be adequate to minimize the total error and to obtain the most accurate model as possible.

3.2.3. Improving the models

Improving the models means increasing its accuracy and predictive power and preventing overfitting, to distinguish noise from data. This process is divided in three tasks:

- **Feature selection:** Consist on removing the variables do not improve gains of the accuracy of the predictions, this allows to reduce computation time and to save storage. This task is performed by many techniques but the aim is to find the variables with more weight in the prediction, by removing the redundant features or by adding features until accuracy is stabilized.
- **Feature transforming:** Is a method to reduce the dimensionality. The main techniques are Principal Components Analysis (PCA)[15], non-negative matrix reduction and factor analysis.
- **Hyperparameter tuning:** Parameter tuning is an iterative process. It begin by setting parameters based on a “best guess” of the outcome. The goal is to find the “best possible” values— those that yield the best model. As the parameters are adjusted and model performance begins to improve, can be seen which parameter settings are effective and which still require tuning.

CHAPTER 4. RESULTS

This chapter shows the results obtained from the machine learning hold-out validation, validations with trajectories generated with PEP and applications on real flight data obtained by DDR2 platform of Eurocontrol.

In commercial aviation, there are two types of distinguished aircraft: the narrow-body and the wide-body. Airbus A318, A319, A320 and A321 are examples of narrow-body aircraft[1]. These airplanes are used for short routes like European continental routes.

The other type of aircraft are the wide-body[2], which covers the A330, A340, A350 and A380 families. The main difference with narrow-body airplanes is the number of passengers to transport. Wide-body airplanes are commonly used for long routes flights or short routes with a high density of passengers in peak hours.

For the study, it has been selected one of the models most used for the airlines operations, the A320-212. The aircraft chosen from the A320 is a short-medium range narrow-body of the Airbus family. Nowadays, Airbus A320 family has over 7.400 aircraft delivered and almost 600 more in backlog orders. Taking back that number, the A320-212 model covers around 4.400, which converts it in the Airbus aircraft model most used by the operators. Thanks to choose the aircraft more used by airlines, will permit to realize a complete study. The high number of real trajectories obtained from DDR2 will provide a good approximation of the flight parameters used to perform a determinate route.

4.1. Experimental Setup

To execute the experiment, it has been chosen the A320-212 aircraft from the Airbus family. The results will give an example for a narrow-body, but it will also be interesting to study a wide-body Airbus model to compare both solutions. In this way, flight performance will variate according to the aircraft model chosen. However, the machine learning algorithm proposed in this final degree project is generic, and could be applied to any aircraft model.

The study for A320 covers distance ranges from 150 up to 3.300 NM and TOMs from 47.000 to 77.000. The total number of flight trajectories generated are 111.100, as it can be shown in the Table B.1, by the equation B.1. But from the total number of trajectories, some have to be deleted because they do not accomplish aircraft performances or characteristics [16]. Some examples of deleted trajectories are those that requires negative payload, or landing weight exceeds the Maximum Landing Weight. In this case, after removing the invalid files the total amount of trajectories used have been 70.319 over 111.100 generated trajectories.

To train the model, we have used Matlab. This program gives access to a Statistics and Machine Learning Toolbox. The application name is Classification Learning, which permits to classify data in many supervised machine learning techniques. Inside the toolbox, the input data must be charged in a very strict format. As explained in the sections before, the output file received from PEP is converted into a table as seen in section 2.3. In the example of the table obtained is Table 2.1, each row represents a different flight while all flight variables are assigned to a different column.

With this table format, the input of the ML is defined, and it allows a better manipulation of all the data required for train the algorithms.

[1] Narrow-body aircraft is an airliner arranged along a single aisle permitting up to 6-abreast seating in a cabin below 4 metres (13 ft) of width.

[2] Wide-body aircraft is a jet airliner having a fuselage wide enough to accommodate two passenger aisles.

Once the data in a table format, there are 3 steps to create your workspace to train the model:

1. **Table selection:** It must be loaded all the data in a table format explained before.
2. **Predictors and Response selection:** From all the data of the table, it can be selected the function variables in the machine learning method. For each variable can be can chosen either *Predictor*, *Response*, or *Do not import*, depending on the relevance in the study.
3. **Validation method:** Once the data is imported, the user could add a validation method to compute the accuracy of the different machine learning techniques. So before to start the training, it can be selected the either if the user wants a *cross-validation* (protects against overfitting by partitioning the data set into folds and estimating the accuracy of each fold), a *hold-out validation* (takes a % of the hole data set to validate the method) or *no validation*.

In our case study, all the variables of the table are predictors except TOM and CI that are responses. As the program does not permit to compute both responses at the same time, it has been studied which is the best way to predict both variables according to their dependence. In the sections below, is explained either both cases with their respective results. To validate the program, it has been chosen the holdout validation taking 10% of the total data set.

4.2. Takeoff mass estimation

First action before to start is to train the model with all machine learning methods in Matlab toolbox to see which one performs better to estimate the TOM. The study will be divided in two cases, one where CI will not be taking into account, and the other where CI will be added as an extra predictor variable to the model. For the former case, 22 predictor variables will be used to estimate the TOM. For the latter case the number of predictor variables will be 23, as CI will act as a known value. A complete table with all methods accuracy can be seen in the Appendix D, Figure D.1.

- **CASE 1:** TOM estimation without the CI value (22 prediction variables).
- **CASE 2:** TOM estimation knowing the CI value (23 prediction variables).

Analysing the accuracy of the TOM estimation, has been observed that some methods present very low accuracy values. To show the best estimation results, the three methods with higher accuracy have been selected for the remainder of the study and the other ones will be discarded because of their lake of trust.

The selection of the best methods will permit to optimize their configuration parameters to be adjusted to our model. The 3 algorithms chosen are:

- **Ensemble Bagged Trees:** Presents an initial accuracy of 77,5% for both cases 1 and 2. Their speed of training is very fast and represents the best accuracy obtained.
- **Fine Gaussian SVM:** Uses Support Vector Machines modeled to a Fine Gaussian function. It is the model which takes longer to estimate TOM values values with an accuracy of 75,6% and 74,8% in case 1 and 2, respectively.

- **Fine KNN:** Uses Nearest Neighbors function. It has the lowest accuracy of both 3, but the time computation for the algorithm is quite fast. It achieves an accuracy of 64,9% in case 1 and 49,5% in case 2.

4.2.1. Dependencies of the variables

A option called multivariables plots allows to obtain a plot with all the variables that are introduced in the machine learning in order to investigate their dependence and correlation.

It is interesting because is a first look for studying if there is an initial dependency between variables. In Figure 4.1 can be seen the colors of the legend used in the multivariables plots, where each color represents a discrete value of TOM. It will be useful to identify dependency between all the predictor variables used in the ML training to find which are the ones that affect more on the ML training and which are redundant (if any).

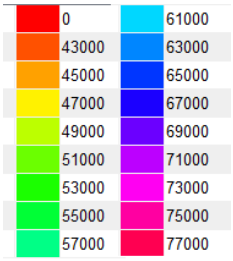
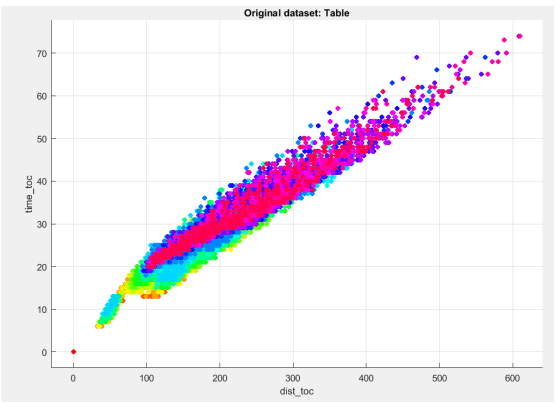
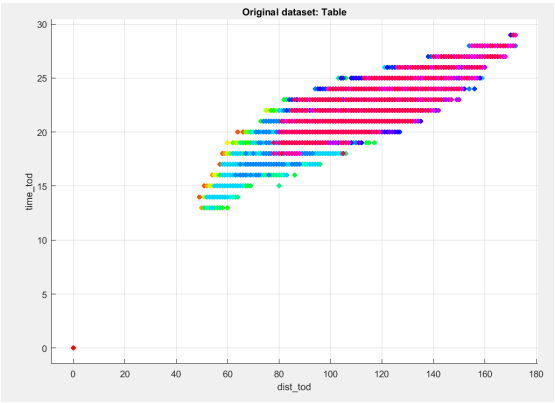


Figure 4.1: Code of colors used for the A320 TOM estimation figures



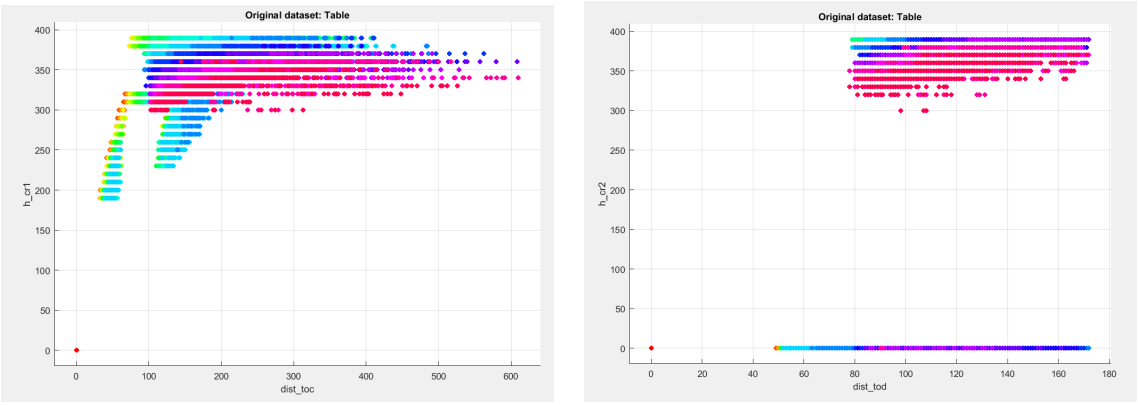
(a) TOM A-320: Distance to TOC vs. Time to TOC



(b) TOM A-320: Distance to TOD vs. Time to TOD

Figure 4.2: TOM A-320: Distance and Time dependence

Figure 4.2 presents the relation between time and distance during the aircraft climb, Top of Climb (TOC) and the descent, Top of Descent (TOD). Figure 4.2(a) shows an increment of TOM implies a longer climb, because both time and distance to the TOC increase due to the fact that the aircraft weights more and the available rate of climb is lower. The same happens in Figure 4.2(b), higher values of TOM implies larger and longer descends. Also, the CI is a delimiter factor during the climb, and the effects of CI during the TOC and TOD will be explained in following section 4.3.1.



(a) TOM A320: Distance to TOC vs. Altitude 1st Cruise

(b) TOM A320: Distance to TOD vs. Altitude 2nd Cruise

Figure 4.3: TOM A320: Distance and Altitude dependence

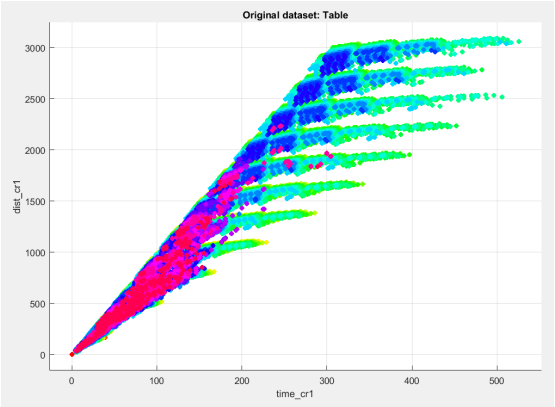
Figure 4.3(a) shows the relation between the distance of the climb phase and the altitude of the first flight level. Similarly, Figure 4.3(b) shows the relation betweenn the distance of the descent phase and the altitude of the second cruise altitude. In the first one, Figure 4.3(a), can be seen that the lowest FL (FL190 to FL290) are reached only for the lower values of TOM. That is because sometimes, for short routes, TOM is limited because of other characteristic weights. An example could be found for short routes. In short flights, the aircraft will not burn much fuel. As all the aircraft have a Maximum Landing Weight (MLW) defined, this weight plus the on-board Trip Fuel (TF) will delimit the maximum TOM available for the flight beeing the MLW and TOM close values.

Also, aircraft always try to achieve its optimal cruise altitude as fast as possible, which is typical very close to it maximum altitude (or ceiling) for the corresponding mass. For high values of TOM and short routes, to reach higher altitudes implies a reduction on cruise time, and cruise represents the design phase for airplanes.

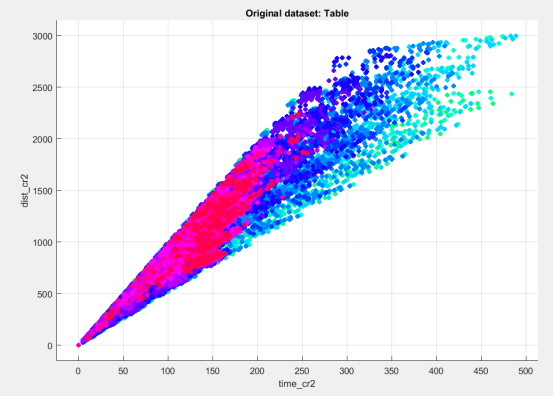
The higher values of TOM are not reaching the higher values of FL, that is because lift has to be equal to the weight of the plane to allow to maintain a constant altitude. In higher altitudes, the density of air is lower and it implies a reduction of lift. As soon as the aircraft losses mass due to fuel burn, it permits to climb to the next FL. Another reason is that for higher values of TOM, lower is the Rate of Climb (ROC) available. The ATC imposes a minimum ROC, that does not allow reaching the highest FL and implies resting in lower FL.

On the other hand, in Figure 4.3(b), is represented the altitude of the second climb versus the TOD. The lowest values of TOM displays altitude 0, that is because a second cruise is not performed. But the airplanes with highest values of TOM that are expected to perform longer routes and realize a second climb. This is because there is a reduction on weight due to fuel burned during the first cruise and as the mass of the aircraft decreases, the optimal altitude increases.

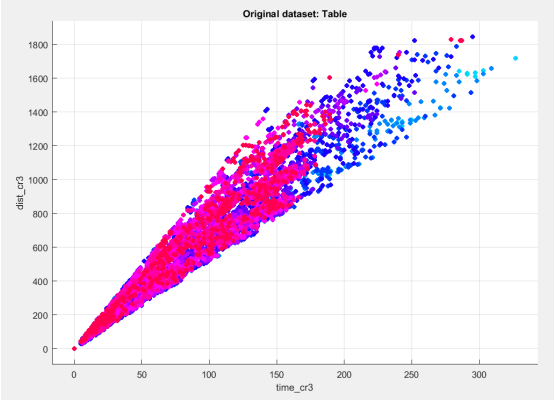
As has been told, the number of cruises performed depends on the distance range values and their range is related with TOM. As larger is range, bigger has to be the value of TOM, because the quantity of fuel to perform the flight has to supply enough power during all the flight.



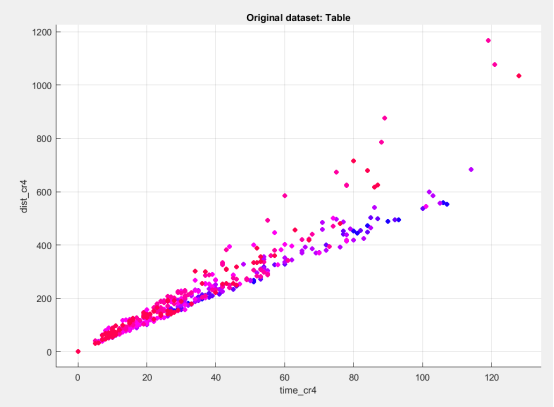
(a) TOM A320: Time vs. Distance 1st Cruise



(b) TOM A320: Time vs. Distance 2nd Cruise



(c) TOM A320: Time vs. Distance 3rd Cruise



(d) TOM A320: Time vs. Distance 4th Cruise

Figure 4.4: TOM A320: Time and Distance in cruise dependence

In Figure 4.4, is presented the time versus distance in each cruise. In a first look, it can be seen that in first cruise, Figure 4.4(a), all the values of TOM are represented, but in the following ones Figures 4.4(b), 4.4(c) and 4.4(d) only the highest values of TOM are represented, and the other ones values are 0, that means that this values are not able to perform other cruises. Another thing that can be seen is that for big values of TOM, the duration and distance in each cruise is lower than other values. The main reason for the distance and time reduction is because these values allow to reach another cruise and is not necessary to stay more time in the lowest FL. The values of the top right in each figure means that has been a long cruise and probably the last one. In that situation, if the pilot try to perform a climb to the following FL it will not be optimal, because it will spend more fuel climbing than resting in the actual FL.

4.2.2. Hold-out validation

Matlab offers an option in each training to show the accuracy of each method. To represent the accuracy for any value of the response required (in this case the value of TOM) there is a graphic called Confusion Matrix.

To obtain this confusion matrix at the first step of the training, when the table is defined containing all the data of the trajectories, a percentage of data is destined to validate the algorithm. In this case a 10% of the overall data will be used to do a hold-out validation of the model. Once the algorithm is trained, the confusion Matrix can be generated. There are two representations: in percentage of

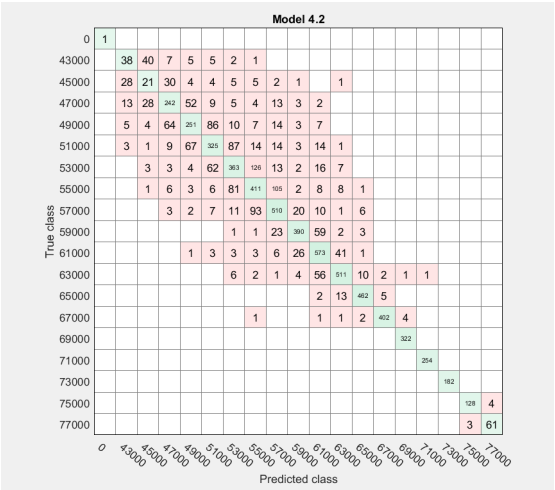
the overall data or in number of trials. The representation in number of trials can be seen in Figure 4.5.

A brief explanation of the following table of hold-out validation multiplot will help to understand the information which it provides.

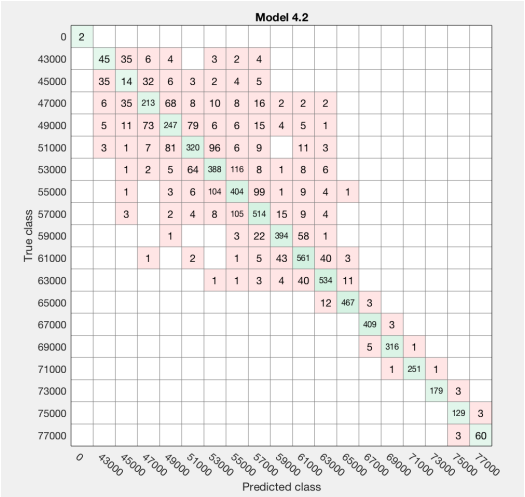
First, numbers in each square of the table corresponds to the relation between the results of CI predictions done (or estimations) respect the true class value of CI. In green are presented the well predicted values, which always corresponds to the diagonal of the table. Error estimations are presented in red and are all the other table squares.

If all the results were over the diagonal, the total accuracy of the model will correspond to 100%, and no error will be presented. As more errors are added, accuracy drops off. With this table can also be studied the precision of the method, which can be extracted if a estimation result is far from their true class value. In this way, the squares far from the diagonal of the table will imply a bad precision of the model.

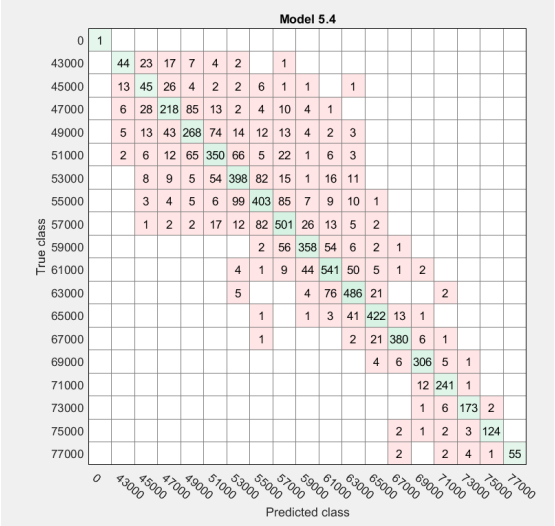
Figure 4.5 shows the best results for the TOM estimation using the A320 model. The one that has shown better results is Figure 4.5(a), which for big values of TOM presents a good accuracy but not for lower ones. The worst case is for TOM equal to 45.000 kg that only has a percentage of hits of 21% from the data trial. The same occurs with the other methods, but the other ones have a worst global result. Globally, the lower values of TOM present worst accuracy results because the high huge of the dispersion is compressed between the initial values of TOM (47.000 - 61.000 kg). The higher values of TOM can be predicted with better accuracy and does not present as much dispersion as lower values.



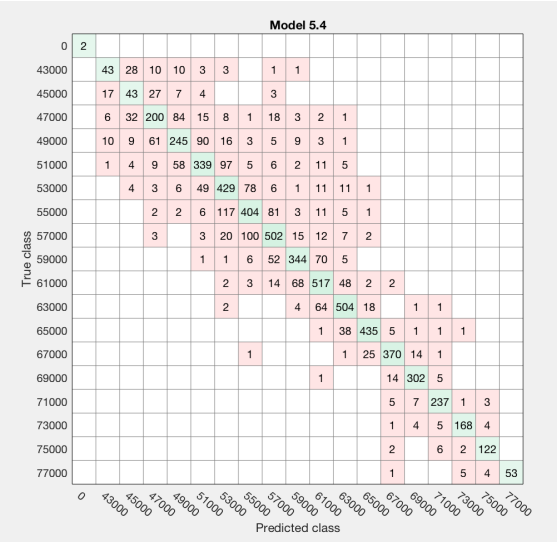
(a) Ensemble Bagged tree (22 variables)



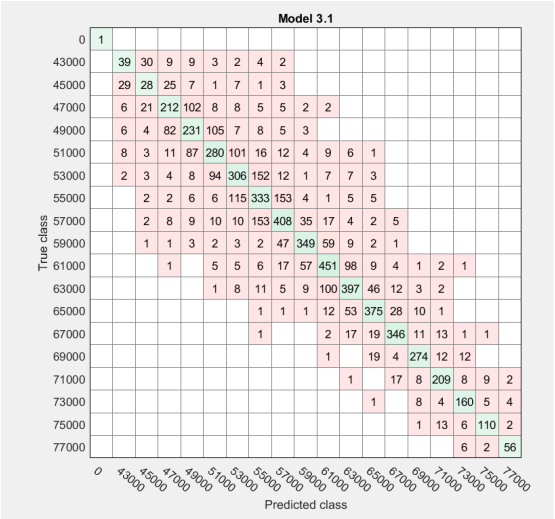
(b) Ensemble Bagged tree (23 variables)



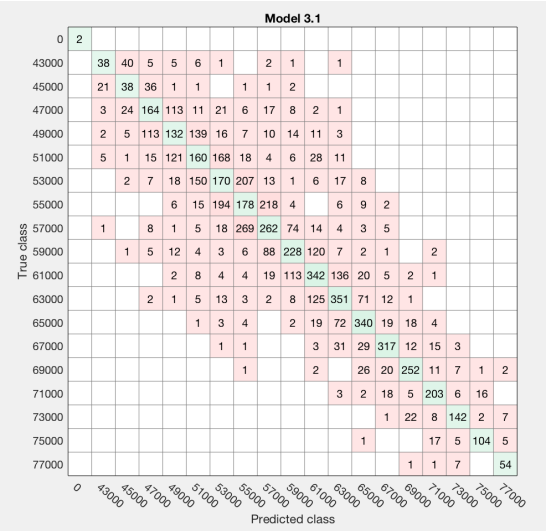
(c) Fine Gaussian SVM (22 variables)



(d) Fine Gaussian SVM (23 variables)



(e) Fine KNN (22 variables)



(f) Fine KNN (23 variables)

Figure 4.5: TOM A320 methods

4.3. Cost Index estimation

As in takeoff mass estimation, all machine learning methods in Matlab toolbox have been tested. In first place, it is computed CI without taking into account the TOM value, so the program uses 22 variables as parameters to determine the estimation of the CI value. After that, the same computation is done but adding TOM as a variable. In this case, TOM value will be an additional predictor variable, so the number of predictor variables will be increased to 23. A complete table with all method accuracies can be seen in the Appendix D, Figure D.2.

- **CASE 1:** CI estimation without the TOM value (22 prediction variables).
- **CASE 2:** CI estimation knowing the TOM value (23 prediction variables).

Some methods present very low accuracy results for CI prediction. This ones will not be taken into account and the study will focus on the three methods that present the best accuracy.

The selection of the best methods will permit to optimize their configuration parameters to be adjusted to our model as has been explained in TOM section. The algorithms chosen are:

- **Bagged decision tree:** Presents an initial accuracy of 76,7% and 77,1% for cases 1 and 2, respectively. Their speed of training is quite fast and the model is able to estimate CI in a pair of minutes.
- **Quadratic SVM:** Uses Support Vector Machines modeled to a Quadratic Kernel function. It has the most accurate response of all possible algorithms with the initial parameter computations with 77,4% for 23 variables. For 22 prediction variables, Quadratic SVM presents an accuracy of 75,4%. In terms of time computation, it is the model which takes longer to estimate CI values.
- **Medium Gaussian SVM:** Also uses Support Vector Machines but approximated to a Medium Gaussian function. It achieve the lower accuracy of both 3, but the time computation for the algorithm is also fast and takes some minutes to estimate CI values.

4.3.1. Dependencies of the variables

As has been done in TOM section, multivariables plot will be used to represent the different predictor variables of the training data set and infer dependencies and redundancies.

Looking for the dependence of the data variables when computing CI, it has been found that distance and time influences CI estimation.

For Figures shown in this Section, it has been used the colors legend shown below, Figure 4.6.

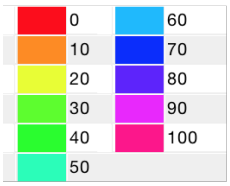
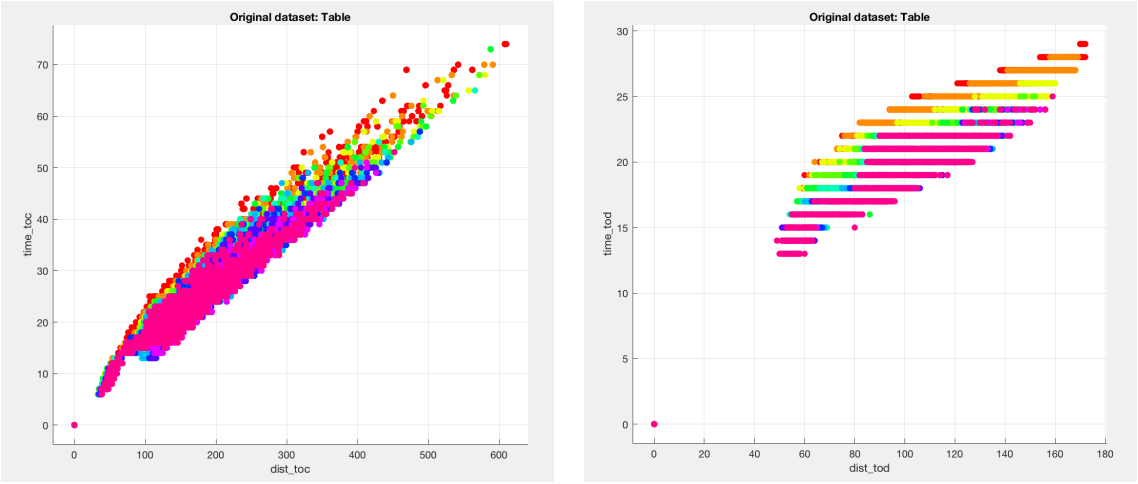


Figure 4.6: Code of colors used for the A320 CI estimation figures

In the following graphs is shown the relation between distance and time when aircraft reach TOC or TOD. Figure 4.7(a) shows that during the climb of an aircraft, the distance to reach the TOC for a fixed amount of time increases with the cost Index. This happens is because CI influences the climb gradient to initial altitude cruise. For lower values of CI, aircraft climbing gradient is the steppest possible because the aircraft aims at reaching the most fuel-efficient altitude as quick as possible, imposing to be the minimum horizontal distance flown by the aircraft. While CI value increases, climbing gradient dwindle and distance is increased gradually with CI value.

In the second graph, Figure 4.7(b), appears a relation between distance and time of descent to reach the TOD variables. For the same reason as before, CI highly influences the gradient of descent from the last altitude cruise. For lower values of CI, the descent gradient corresponds to the minimum angle of descent. Given a determine time of TOD, this factor implies to fly the maximum horizontal distance possible. So, as CI value increases, the descent slope becomes more steep making the distance progressively decrease.



(a) CI A320: Distance to TOC vs. Time to TOC (b) CI A320: Distance to TOD vs. Time to TOD

Figure 4.7: CI A320: Distance and Time dependence

The climb to TOC slope is one of the effects of the CI value. In Figure 4.8, can be seen this effect, where for lower values of CI the slope is steeper than for the higher ones, which pretends to arrive to TOC in less time than before. While for values from 0 to 50, the slope is clearly separable, but from values of CI equal to 60 or more, the slope does not present relevant changes on the slope. That will imply that the prediction of CI over 60 will not be as accurate as for the lower values of CI. In the following plot is represented the distance TOC versus the first cruise altitude in FL. All the samples have been taken for the same value of TOM of 65.000 kg and the first FL of 37.000 ft.

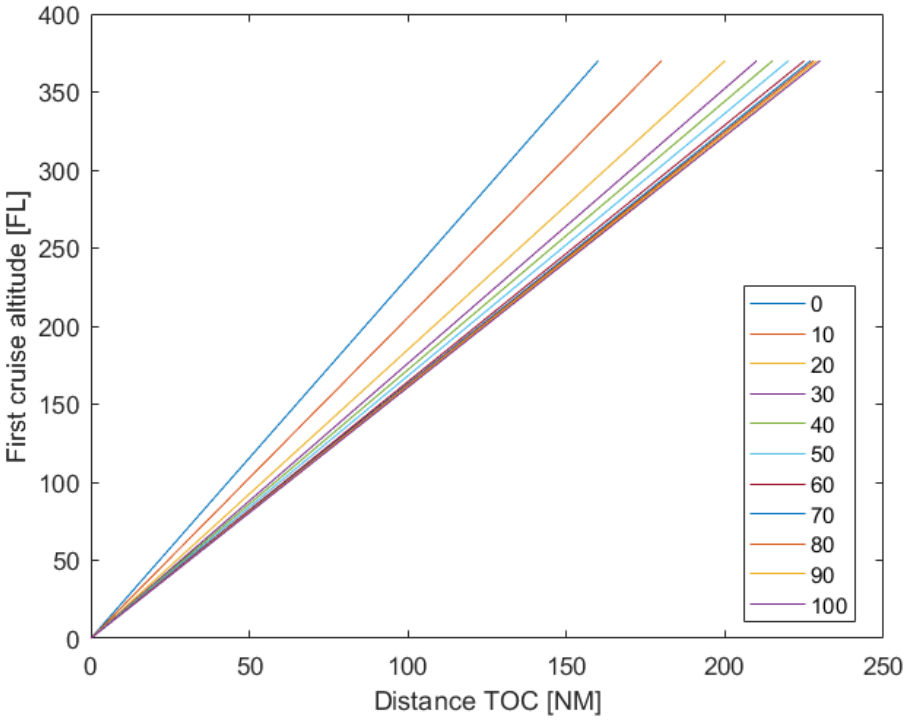
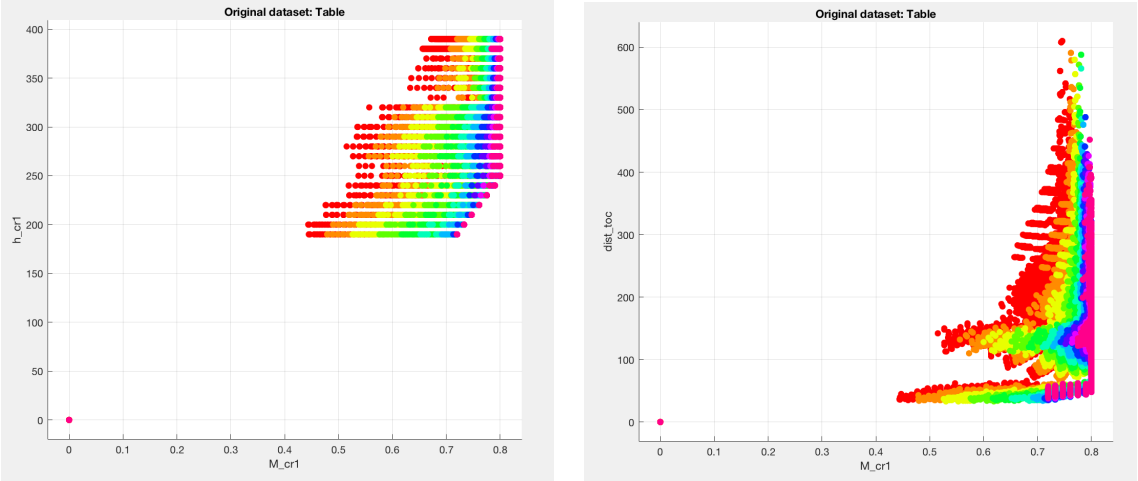


Figure 4.8: Climb performance, TOW=65.000 kg

Another parameter which affects CI estimation is Mach Number. The following plots present the dependence on Mach Number in cruise respect FL and horizontal distance to TOC. Figure 4.9(a) shows the relationship between mach number and cruise altitude for different Cost Index. One important parameter to remember is that time cost appears in the numerator of the CI ratio. Considering a fixed value for cruise altitude, for low values of CI, time cost relevance will be almost null. This fact will imply Mach Number at cruise to take low values too. The case when CI increases, time cost gains importance, so Mach Number needs to increase to higher values.



(a) CI A320: Mach Cruise vs. FL

(b) CI A320: Mach Cruise vs. Distance to TOC

Figure 4.9: CI A320: Mach Cruise dependence

In Figure 4.9(b), for the same reasons as before, Mach Number in cruise is directly related to time cost operation. For a given value of distance to TOC, low values of CI will imply lower Mach Number values. This is because when time cost are low, time influence almost disappears so Mach Number could decrease to values such us 0,5 or 0,6. Once time gains importance, Mach Number has to be raised until it reaches the maximum operating Mach Number.

To show the relationship between the Cost Index and the cruise Mach, values are represented in a box-and-whisker plot which permits to show results in a interquartile range (abbreviated IQR). The IQR tells how spread out the middle values are. It can also be used to tell when some of the other values are too far from the central value. Whickers are used for values does not enter to the quartiles defined by IQR but are not too far away points. These points further than whickers are called outliers, because they lie outside the range in which values are expect to be.

On each box, the central mark indicates the median [3] Mach velocity, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The default value for whiskers corresponds to approximately the 99.3 percent coverage if the data are normally distributed. As Figure 4.10 shows, the plotted whisker is extended to the adjacent values depending on the IQR amplitude, which is the most extreme data value that is not an outlier. Whiskers are always symmetric respect the IQR box, except for the cases where values extend to the limits of the study or rather there is no value on the whisker zone. Further points or outliers are plotted individually using the '+' symbol.

[3] Median value: is the value separating the higher half of a data sample from the lower half.

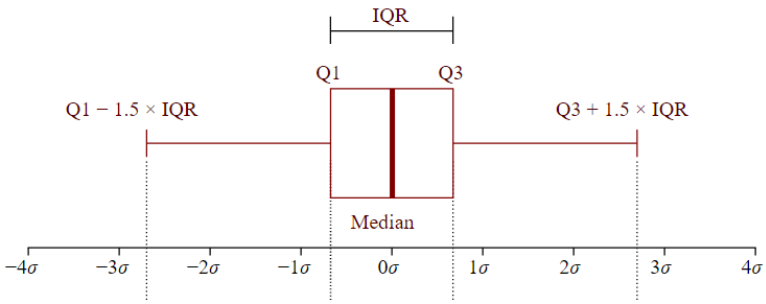


Figure 4.10: Box-and-whisker model

Figure 4.11 shows the relationship between the Cost index and the cruise Mach. For big CI values, the number of Mach is increased, that shows the CI effects on the flight velocity, and the prioritization of time against fuel consumption. Whiskers permit to say that as CI values increase, dispersion on Mach velocities decrease. The deviation of the median value, can be consequence of the wind, ISA and range, because the only filtered applied to obtain this boxplot has been the value of TOW equal to 65.000 kg. Also, CI values from 50 and up some some Mach values are equal to MMO and finally for CI 100 there is no IQR because all the values of Mach are MMO, that means that the airplane is flying a maximum velocity.

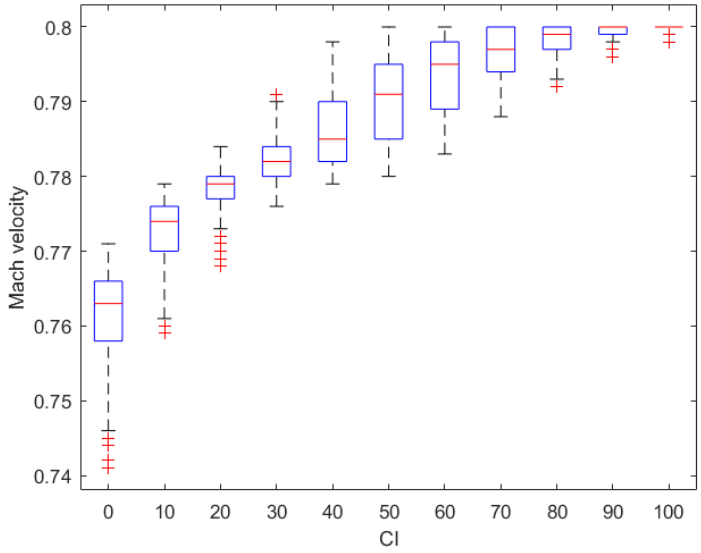


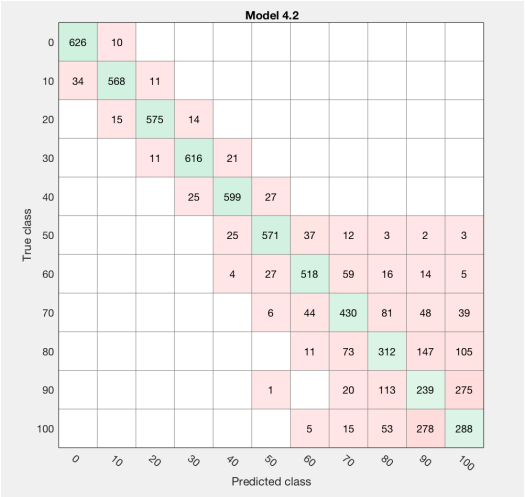
Figure 4.11: Mach dependency for CI, TOW=65.000 kg

4.3.2. Hold-out validation

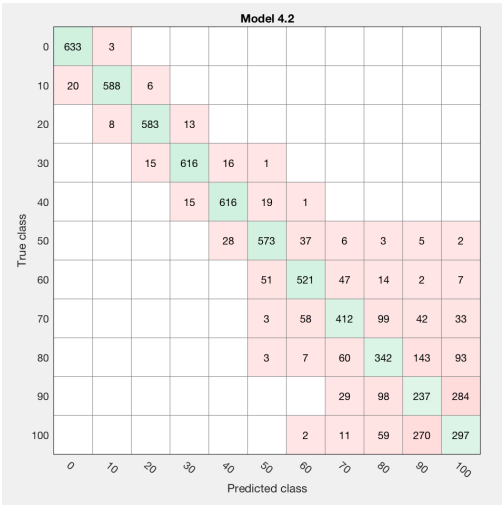
As it has been seen, in the section before, accuracy changes depending on the machine learning algorithm used to predict. Now, in this hold-out validation multiplot, is shown the number of predicted CI values for each model. For the hold-out validation has been taken the 3 model predictors with the higher accuracy. This models will be studied in terms of accuracy, precision or even tendencies on CI estimations.

The hold-out validation Figure for the CI is analogous to that of the TOM. The diagonal is green when the value of the prediction is the same as the true value, which means that the algorithm has to predicted right. The other squares in red mean that the prediction value does not correspond to the true value.

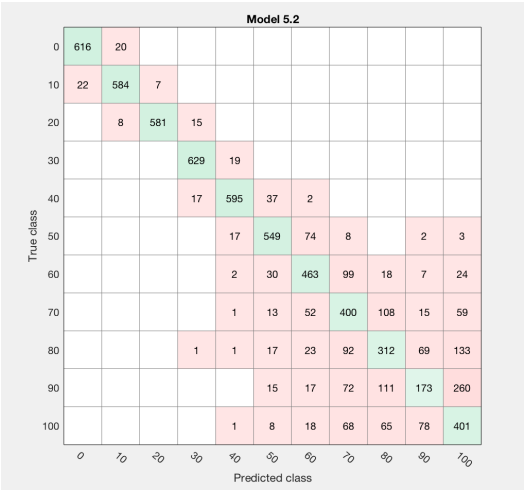
The way to present the hold-out validation table is using the number of estimations for each class prediction, following the method used in TOM. Hold-out validation multiplot presents the three best models for 22 and 23 prediction variables:



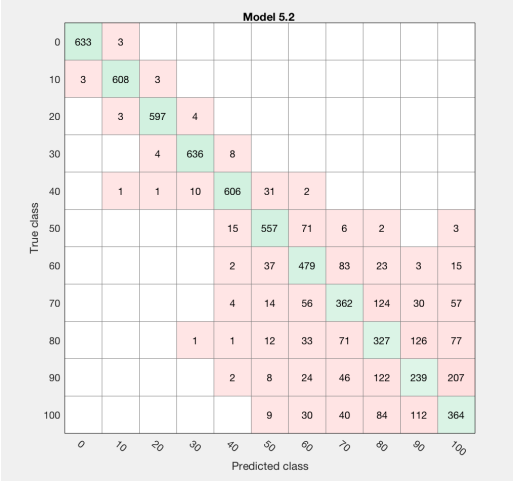
(a) Bagged decision tree (22 variables)



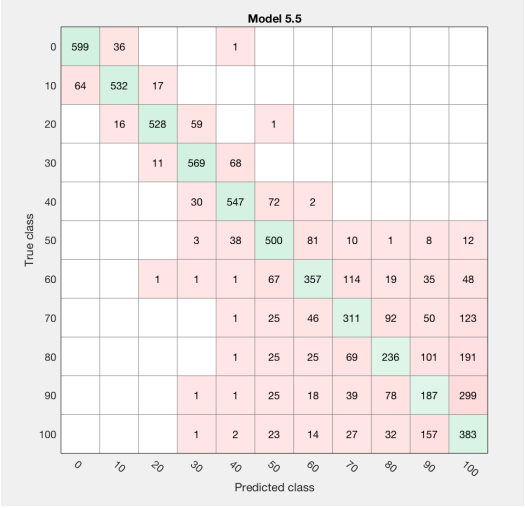
(b) Bagged decision trees (23 variables)



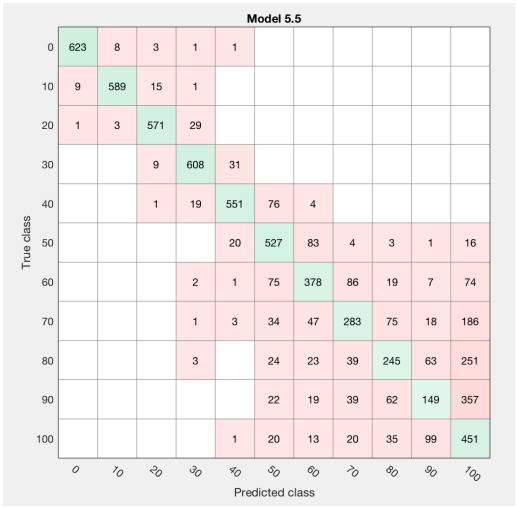
(c) Quadratic SVM (22 variables)



(d) Quadratic SVM (23 variables)



(e) Medium Gaussian SVM (22 variables)



(f) Medium Gaussian SVM (23 variables)

Figure 4.12: CI A320 methods

In this case, from the graphs and accuracy results, can be extracted that with 23 prediction variables the model is able to estimate almost the same as 22 prediction variables. It can be observed that adding the TOM as predictor variable does not improve the estimation significantly. Even so, some CI estimated errors from 22 prediction variables could be slightly reduced.

Another point to comment is how CI estimation accuracy depend on the CI value predicted. While for low values of CI, the estimation accuracy is all over 90%, when CI passes over 50 the model is not able to estimate in such precision. For high values of CI, accuracy drops off significantly because for higher values of CI Mach Number trends to MMO and all the trajectories became very similar regardless of the CI value.

Paying attention to the best algorithms used, bagged decision tree seems the most precise model for CI estimation. Estimations for low CI never deviates more than 10% from the true class value, even thought for high values precision is not so good and range error is increased. For both SVM models, Quadratic and Medium Gaussian, present very low precision for high CI values.

In conclusion, for all algorithms, accuracy is higher for lower values of CI. Also, precision is good because estimation errors are all near the diagonal of the table. When CI value increases, precision and accuracy decreases notably.

4.4. Method Selection

After training the data set with different ML methods, is time to decide if the CI and TOM estimations are better predicted separately or step by step, and which is the best method . With ML Toolbox of Matlab, the program is only able to calculate a prediction for each execution of the algorithm. It is not possible to predict two responses at the same time, that means, it can not be predicted the value of CI and TOM at the same time. Thus, two separate runs would be required to estimate the CI and TOM values of a certain trajectory.

Now, the different prediction combinations will be studied to find the best total accuracy for both estimations. For the study, it will only be considered the machine learning methods with best accuracy values.

The obtained accuracy results for both CI and TOM predictions with 22 and 23 variables are the followings:

	CI	**TOM**
22 variables	76,7 %	77,5 %
23 variables	77,4 %	77,5 %

Table 4.1: Initial accuracies for A320

Table 4.1, shows the results of CI and TOM estimations with 22 prediction variables are very close to the results for 23 variables. All this algorithms are tested with the default configuration parameters of each model, but ML Toolbox allow an advanced option where initial parameters could be modified. This will permit to improve the model accuracy and get better response estimations.

One possible method is to compute the accuracy for CI and TOM individually. Only taking the accuracy of the 22 prediction variables method, the accuracy of each variable will be executed in an independent way. In this case, no error will be accumulated and the accuracy of the estimation will be directly the same as computed with 22 variables.

Another issue to be considered is whether CI and TOM estimations are dependent or not. If we start from an initial scenario where both CI and TOM are not known, estimations for 23 variables will imply a previous estimation of the other variable. This previous estimation will introduce a certain accuracy error which will be accumulated to the actual estimation error of the remaining variable. So, both results for 23 prediction variables will accumulate the accuracy error from the previous estimation.

Finally, this study is based on an iterative method. As the best accuracy results are obtained from 23 prediction variables, it will be interesting to consider a method that implies both variables simultaneously without the accumulative error. This method tries to minimize the error by consecutive loops, where the values of CI and TOM are computed iteratively until results converge to the previous step.

In this method, the first interaction computes the TOM with 23 variables of input, but the variable corresponding to CI is empty because it is unknown. Then, knowing the TOM predicted value, the CI is computed. The following steps consists on the same process until the prediction of the last step coincides with the previous one.

In the case of the A320 has been observed that the best algorithm for TOM estimation is Ensemble Bagged Tree with an accuracy of 77,5%. For CI estimation, also Ensemble Bagged Tree appears as the best estimation algorithm with 77,4% of accuracy. This results have been obtained for initial parameters but the advanced options permit the number of learners to be modified, which will suppose an accuracy improvement.

4.4.1. Ensemble Bagged Tree Optimization

As it has been seen, the best way to compute CI and TOM is by means of an iterative method. Despite we obtain an accumulative error due to computing both predictions together, the iterative method permits to minimize that error and adjust the predictions increasing the accuracy in each loop.

Once chosen the method to be used during the study, the following step is to optimize the algorithm parameters to acquire the maximum accuracy for the estimation. There are four parameters which influence the accuracy for Ensemble Bagged Tree:

- **Maximum number of splits.**

  As the number of splits grows, the complexity of the tree increases with it. A correct number of splits will permit the model to improve the accuracy, but you must be careful because a wrong value of number of splits could cause overfitting. The main way to choose the best tree depth for the trees in the ensemble is using trial and error method. To see their influence on accuracy, all parameters have been defined as constant values while changing the maximum number of splits value.

- **Number of learners.**

  The predictive power is increased as the number of learners does. Accuracy improves at the cost of increase in time processing, complexity and memory usage. To use many learners can produce high accuracy, but can be time consuming to fit. Normally, to have a model with a high predictive power is needed a few hundreds of learners.

- **Learning rate.**

  The value is set by default at 0,1 and cannot be modified in Matlab toolbox. If you set the learning rate to less than 1, the ensemble requires more learning iterations but often achieves better accuracy.

- **Subspace dimension.**

  For subspace ensembles, consists on specify the number of predictors to sample in each learner. Matlab chooses a random subset of the predictors for each learner. The subsets chosen by different learners are independent. For Bag Ensemble Method, subspace dimensions is 1 and cannot be modified. If the user wants to set a determinate number of subspace dimensions, the method will change from Bag to Subspace. This method is mainly used when the study requires many predictors.

To find the best accuracy parameters, the ensemble bagged tree has been executed with different values of maximum number of splits and number of learners. Learning rate and subspace dimension have been defined as constant, with values of 0,1 and 1, respectively.

After using trial and error method, it is obtained the best accuracy model for ensemble bagged tree. The study have been realized for both CI and TOM with the iterative method, obtaining the following results:

- **TOM:** To obtain the best accuracy model, the values of Maximum Number of Splits and Number of Learners has been changed in many tests. Finally the best results have been obtained with the combination of: 26 Maximum Number of Splits and 150 Number of Learners. The prediction of TOM accuracy is 78,6%.

- **CI:** After trial and error method, it is obtained the best accuracy model with the combination of: 14 Maximum Number of Splits and 150 Number of Learners. The prediction of TOM accuracy is 78,2%.

4.5. Variables sensibility

After the optimization of the best Machine Learning method Ensemble Bagged Trees, as has been seen in section above 4.4.1. is time to compute the variables sensibility. In this case, it will be computed by training the model with the optimal parameters but deleting some of the inputs variables. Then, the analysis of the increments or decrements of the accuracy obtained will determine the

importance of each variable in the model. In Figure 4.13 is represented the block diagram of the process, it is an interative process where in each step is deleted one variable as shown in Table 4.2.

Figure 4.13: Block diagram for variables sensibility

	A320	
Variables	**CI**	**TOM**
Reference Accuracy	*78,2%*	*78,6%*
Velocity (Mach)	*-23,8%*	*-3,9%*
Total Time & Distance	*+2,0%*	+3.7%
Time & Distance TOC	*-7,5%*	*-14,9%*
Time & Distance TOD	*-1,0%*	*-6,1%*
Cruise Altitudes	*-0,2%*	*-7,8%*
TOM/CI	*-1,6%*	*-1,0%*
Adding Wind & ΔISA	*+2,4%*	*+2,7%*

Table 4.2: Affectation in increment by deleting variables

For the study, it has been taken as the reference value the optimal accuracy obtained from the optimization. As can bee seen in section 4.4. for the algorithm of computation of CI is 78,2% and for TOM is 78,6%. From here, the difference obtained from the reference accuracy and the new training accuracy by deleting input variables, such us velocity, time, distance and CI and TOM is presented in the Table 4.2.

From Table 4.2 can be observed that from deleting total time and total distance, response predictions improves its accuracy by 3%. The reason of that improvement is that total time and distance are redundant data. In all the flight phases (climb for TOC,Cruises and descent for TOD) distances and time are considered. If it is needed to obtain the totals, it could be computed by the sum of the different phases, so that input variables give the information by duplicate. Also, distance and time totals generate overfitting to the model. Many CI and TOM could belong to different trajectories. This fact would increment the complexity of the algorithm and would have a negative repercussion on the final predictions. By deleting both variables, the accuracy of the training increases and the errors on predictions would be reduced.

For the rest of the variables, their removal have a negative repercussion in the model accuracy, but not with the same impact for CI and TOM estimation. For example, velocity have a highest effect in CI, because in function of the CI selected in FMS if time is minimized, velocity increases. But there is no such relation between the weight of the aircraft and the velocity, because the reduction of the accuracy is less than 4%. On the other hand, the variables related with trajectory performance like altitudes, and distance (TOC and TOD) have more affectation in TOM, because the aircraft weight affects in flight performance, reducing the cruise altitudes available or hindering the climb.

Additionally, it has been interesting to consider wind and ISA deviation. The meteorological conditions have a relevant impact on trajectory path. For a given value of distance the trip time could

vary depending on the wind and ISA affectation . Wind and ΔISA are parameters which has to be obtained depending from a weather observation. This data has been considered for the trajectories generation, but initially, there were not considered in the ML because from the real surveillance application of DDR2 data, this information required for the input is not available.

The fact of adding wind and ΔISA also improve the model accuracy. Both variables contribute with relevant information for the model and permits to adjust the algorithm and to be more precise. Affectation of adding both prediction variables are even the same, an increment of 2,4% for CI, and 2,7% for TOM.

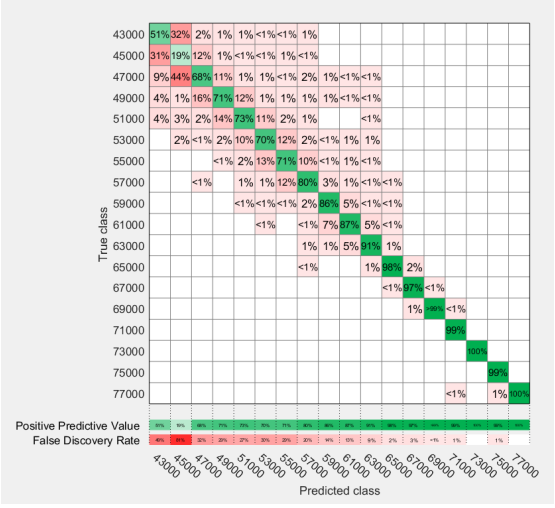
Finally, after analysing the affectation of each variable, it is seen the best accuracy model is obtained by deleting total time and distance and adding wind and ΔISA. The former variables generate overfitting and the algorithm has redundant information by duplicate and gets lost. The conclusion is to eliminate total time and distance variables from the following studies, so accuracy improves as seen in section 4.4.1. The addition of the later variables will depend on the application we are working on. Depending on the application, if wind and ΔISA are known parameters, will be used in the algorithm to predict CI and TOM values. If the application does not contain that information, the algorithm used will be the previous model explained, without taking this variables in algorithm for the prediction.

	A320	
Variations on model variables	**CI**	**TOM**
Deleting Total Time and Distance	*80,2%*	*82,3%*
Adding Wind & ΔISA, without Total Time and Distance	*82,7%*	*87,0%*

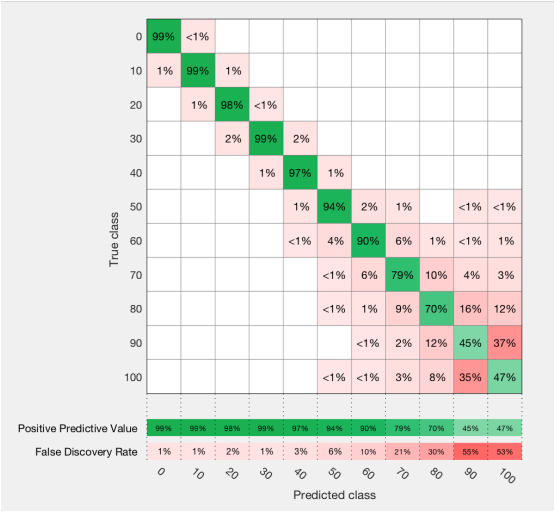
Table 4.3: Accuracy models

From this models, accuracy is improved in the critical prediction values. For CI, accuracy for high values is increased and is able to predict with better reliability, while for TOM, prediction for lower values are notably improved. The confusion matrices 4.14 show the accuracy for all the different values of CI and TOM.

(a) CI A320 Deleting Total Time and Distance



(b) TOM A320 Deleting Total Time and Distance



(c) CI A320 With Wind and ΔISA

(d) TOM A320 With Wind and ΔISA

Figure 4.14: Variations on model variables

4.5.1. Study of the variables with noise

In real life, data have an implicit noise due to errors in measurements and calibration. As samples will have some noise, a deep analysis must be done to know the noise affectation on the model and the final prediction accuracy.

To see if the ML algorithm is robust and capable of accurately predict the CI and TOM values, a noise study on the different variables will be done. On the section 4.5. variables were eliminated from the ML study to see their affectation on the prediction model. Now, all variables will be able for the study, but their values will be perturbed with a random noise to see how noise degrades the prediction accuracy. Matlab allows to define the probability distribution affected by the random measures. In this project, a normal distribution has been selected because the most typical example of noise.

The noise sensitivity analysis will performed for each predictor variable independently, in order to isolate their effects. In the Table 4.4, it has been selected the variables to be modified and their possible range of variation due to noise on the samples. The data set used for the analysis is the same as the input of ML training but adding a random value between the variation range for each variable. To choose the noise variation range, it has been assumed to have the worst case possible, with extreme noise on the samples.

To compute the final accuracy of the new data set, it has been used the prediction model from ML algorithm found in Figure 4.14(c) and 4.14(d). While wind and ΔISA will be added to the study, total time and distance of the flights will be removed and not taking into account in the analysis because it permits to improve the accuracy of the model (82,7% for CI prediction and 87,0% for TOM prediction).

Variables		Minimum	Maximum	Variation
Altitude [ft]	h	h-500ft	h+500ft	1000ft
Mach Number	M	M*0.96	M*1.04	8%
Wind and ISA deviation	w	w-15kt	w+15kt	30kt
[kt] and [ΔISA]	ΔISA	ΔISA-10	ΔISA+10	$\Delta 20$
Cruise time and distance	t_cr	t_cr-2.5min	t_cr+2.5min	5min
[min] and [Nm]	d_cr	d_cr-4Nm	d_cr+4Nm	8Nm
TOC time and distance	t_toc	t_toc-2.5min	t_toc+2.5min	5min
[min] and [Nm]	d_toc	d_toc-4Nm	d_toc+4Nm	8Nm
TOD time and distance	t_tod	t_tod-2.5min	t_tod+2.5min	5min
[min] and [Nm]	d_tod	d_tod-4Nm	d_tod+4Nm	8Nm

Table 4.4: Noise variation on variables

Once the random noise is introduced in the data set, the program is executed to predict CI and TOM in an iterative way. If the noise was equal to 0, all prediction will coincide with the known value of CI and TOM. This is because the data set will be equal to the training input of the ML model and accuracy will be 100%. When noise is introduced, sample values change from the input data set, so the training model would not correspond to the new sample with noise.

The accuracy has been computed by the comparison of the prediction value and the class value of the input data set. So, the final accuracy is assumed as the percentage of the number of hits on the prediction respect the total number of predictions.

	A320	
Variables	**CI**	**TOM**
Altitude	*94,88%*	*99,96%*
Mach Number	*39,16%*	*82,93%*
Wind and ISA deviation	*93,54%*	*93,28%*
Cruise time and distance	*93,29%*	99,33%
TOC time and distance	*85,52%*	*80,60%*
TOD time and distance	*80,50%*	*93,12%*

Table 4.5: Accuracy after noise affectation

From Table 4.5, can be seen that TOM predictions are more accurate than CI predictions. For CI, the main variables affected by the noise are the Mach Number and TOC/TOD times and distances. Clearly, the main parameter to predict CI is the Mach Number, which with a minimum variation of it, the accuracy of the prediction would dramatically decline. In this case, with a noise variation of 8% on the Mach Number, accuracy would drop off until a 39,16%. Also, affectation of noise is present in TOC/TOD times and distances. Both are two phases of flight were gives very relevant information to compute the CI value, and the accuracy would be 85,52% and 80,50%, respectively. For the rest of variables, accuracy on the predictions remains over the 90% which is still a reliable result.

Looking at the TOM predictions, there are only two variables which accuracy is below 90%. When noise is added on the Mach Number, the affectation on TOM prediction also decrease to 82,93%. But the most relevant noise affectation appears in TOC time and distance. A crucial phase of flight to determine the point which determines the maximum TOC of the aircraft is the climb phase. When noise is added to time and distance to reach the TOC, the accuracy of the prediction reduces to 80,60%.

In this case, accuracy is not enough to determine if the noise effect on the predicted results deviate more or less from the true value. To show the error committed, it have been computed the Mean Absolute Error (MAE) and the Standard Deviation (SD). Table 4.6 shows that for CI in Mach Number noise, MAE corresponds to $\pm$ 18,57 with a SD of 21,10. CI accuracy present a low value of 39,16%, and with the error study, can be computed that the deviation of CI could change almost 37 units from the true value. In the case of TOM the most affected variables are Mach Number and Wind & ISA deviation. Mach Number presents a MAE equal to $\pm$ 892,20 with a SD of 1.926,90, so the error is about the order of 2.000kg. Wind & ISA deviation shows less error, with a MAE of $\pm$ 407,33 and a SD of 1.731,30.

	CI		TOM	
Variables	**MAE**	**SD**	**MAE**	**SD**
Altitude	$\pm 0,66$	3,14	$\pm 1,05$	68,53
Mach Number	$\pm 18,57$	21,10	$\pm 892,20$	1.926,90
Wind & ISA deviation	$\pm 0,66$	3,14	$\pm 407,33$	1.731,30
Cruise time and distance	$\pm 0,87$	3,78	$\pm 15,42$	203,49
TOC time and distance	$\pm 1,93$	6,17	$\pm 576,55$	1.449,70
TOD time and distance	$\pm 2,08$	7,36	$\pm 183,19$	773,06

Table 4.6: MAE & SD for noise analysis

To avoid errors due to noise, the data set samples must be accurate. Noise effect on the variable values should be reduced as much as possible to obtain the best predictions as possible. So, as less noise have the data samples, better predictions could be done with the ML algorithm. In the case of CI, Mach Number and TOC and TOD values must be reliable to obtain an accurate result. For TOM predictions, noise does not affect as much as in CI, but also errors on Mach Number and TOC time and distance would misfit the data and generate false results. In some situations not only noise will be the problem, it is possible that due to ATC orders or pilot actions could generate deviations from the optimal vertical trajectory. This deviations could be assumed as noise because is a variation respect the optimal vertical profile.

4.6. Model validation

Sometimes the model that shows the best accuracy by Matlab does not imply that is the best solution because the algorithm has been designed for training data and could present divergence between training and test data. For that reason, one validation activities have been carried out. The algorithm trained with PEP data has been used to predict the CI and TOM of other trajectories generated with PEP but not included in the training data set.

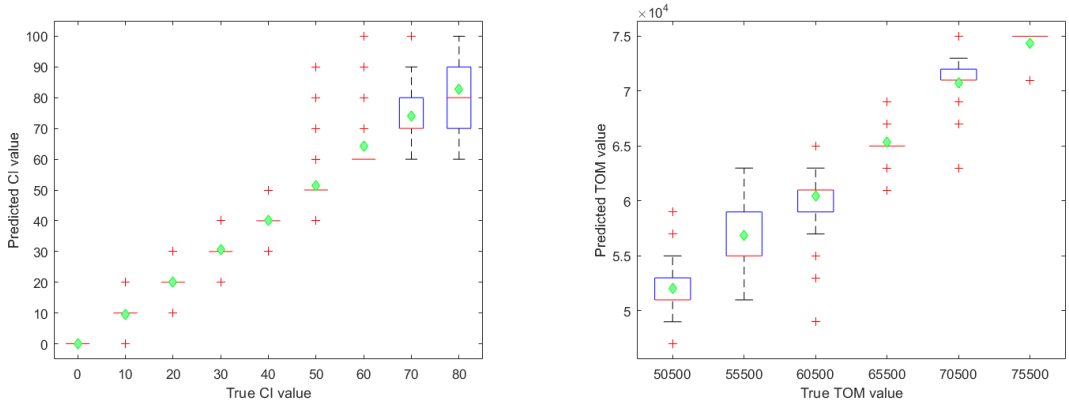
4.6.1. Validation with PEP

A set of 1,536 trajectories has been generated with PEP for different ranges, CI and TOM. These trajectories will be introduced to the trained algorithm to predict their CI and TOM. Once obtained this predicted values, they will be compared to the true values to examine the accuracy.

To try to improve the algorithm predictions, the parameters of the predictions will be changed to maximize the accuracy, because the initial accuracy do not imply that is the optimal one for test data, as explained previously on section 3.9. As has been seen, the fact of realizing a training with less data does not downturn the accuracy. Increasing the complexity of the model does not

ensure an improvement of the accuracy. The complexity of the algorithm is related to the number of branches of the complex tree and what can be the best result for training does not imply to be the best prediction for test data.

First of all, a validation of the best algorithm obtained in previous sections is going to be analyzed. This algorithm does not takes into account the wind and ISA deviation because the final objective of this algorithm is the implementation on real cases where wind and ISA conditions are not available in DDR2 platform.



(a) CI validation with PEP data.

(b) TOM validation with PEP data.

Figure 4.15: A320 validation with PEP data

Figure 4.15 shows that in the first one where the CI is estimated, is verified that for lower values of the CI the prediction is better than for the higher ones. The ideal case of 100% of accuracy would be represented as a straight line of 45^o of slope. Also, it is interesting to see that the mean values tends to be near the straight line of maximum accuracy, so the results seems to hit the CI predictions. On the other figure of TOM, is corroborated that the worst accuracy is obtained for the lower values of TOM. In this case, the mean values of TOM predictions are not so close to the 45 slope line of the 100% of accuracy, so the predictions hits are not as good as CI. For this initial test of validation, the percentage of hits for CI prediction has been of 81,77% and the one for TOM 67,71%.

Once arrived at this stage, it is interesting to make a study of the overfit of the algorithm, because it exists the possibility that our algorithm is optimized for training data but not for test data, as seen in section 3.2.2. For this reason, using this test data of 1.536 trajectories and our algorithm, by changing the complexity parameters will be obtained the optimization for data test. The parameters that affects on the algorithm complexity for the Ensemble Bagged Trees are two, the number of splits and the number of learners. By changing the value of them, as can be seen in Appendix E, it can be performed different tests and get the percentage of hits for each algorithm.

Finally, we can conclude that the affectation of changing the complexity parameters does not considerably change the number of hits per both predictors. As it is not so relevant, for further studies it will only be considered the optimal algorithm found in training section. From the prediction test results in Appendix E is shown that the accuracy from the training data (given by Matlab) is not equal to the obtained in the validation process. While in CI the hits are almost the same value as the accuracy of training, that does not succeed with TOM. For this reason, it is necessary to add a validation process where new data is tested to check the classification learner of Matlab results. Another aspect of interest is that in Ensemble Bagged Trees, the changes on number of learner and number of splits, have no several impact on the test data due to a variance of $\pm 1\%$ for both CI and TOM predictions.

4.7. Application with real surveillance data

The algorithms have been trained, optimized and validated as has been seen in the upwards sections. In this section real flight data is going to be used to compute the CI and TOM of flights that have flown in European Airspace, to realize this section the data required has been taken from DDR2, that is an online platform of Eurocontrol.

4.7.1. Data extraction

DDR2 platform contains all the historical data of European flights but presents an option to filter the data as can be seen in Figure 4.16.

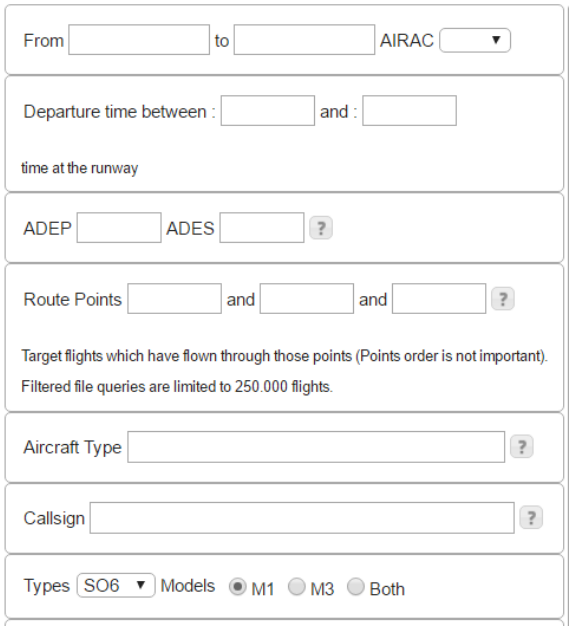


Figure 4.16: Filtering menu of DD2

In the filter can be selected the day of the study filling the boxes From... to..., and the period of hours in the following ones (Departure time between: ... and: ...).

ADEP and ADES means the region where the plane departures and where arrives, respectively. Can be selected a area (i.e. LFP = Paris area) or an airport (i.e.EGLL = London Heathrow), this is useful to apply an study of a certain route.

Also, the study of a certain route can be performed with the box of Callsign, the callsign is the code of a certain flight, this contains the ICAO letters of the airline and 4 numbers to describe the route (i.e. ABC1234, where ABC correspond to OACI code of the airline and 1234 the flight number).

The box with Route Points, is to filter by the airplanes that perform a flight that have passed thought the selected waypoints. This section is not going to be used because all the studies only contemplates the origin and the destination.

The box of Aircraft type will be the most useful for this case study. With this filter can be filled with A320 to demand the all flights that has been perform with Airbus-320 model.

Finally the last boxes, Type S06 M1 M3 Both, is to select the type of model trajectory. While M1 corresponds to the last-filed flight plan, M3 is the actual radar-tracked flight like has been explained in 1.7. In this case we are going to use M3 files because are the actual trajectories which have been

computed by the FMS on-board with the corresponding updates and regulations. Once has been extracted the files to study this contains a list of segments of the flights. Each segments contains data about position, times, callsign, aircraft model, and length of the segment.

An algorithm has been developed in Matlab to perform the following operation:

1. Classify all the lines of the files in function of the flight to which it belongs.
2. Order the data of each flight in function of the time, to have all the segments in temporary order.
3. Group the segments in flight actions: Climb, cruise or descend. Also, the data that are wrong has to be deleted, for example, a level-off[4].
4. In each of the previous groups the relevant data has to be extracted with the format of Tabale 2.1.

This data do not present the velocity in Mach Number but in ground speed. In the algorithm trained the Mach velocity of cruise is an input required, for that we are going to perform the following calculation of the the Mach, with the following assumptions.

- The flight is perform with ISA Atmosphere.
- There is no wind, there is calm atmosphere.

$$M = \frac{V}{V_{Sound}} \tag{4.1}$$

Where:

$$M \quad = \quad Mach \quad Velocity$$

$$V = \frac{Segment \quad Length}{Time} \tag{4.2}$$

$$V_{Sound} = \sqrt{\frac{\gamma RT}{M}} \tag{4.3}$$

In equation 4.3: γ is the adiabatic index, γ=1.4; R is the constant gas for dry air, R=8.314 [*J/(K·mol)*]; M is the molar weight of air, M=28.95·10-3 [*kg/mol*]; and T=Temperature(altitude) [*K*] applying ISA Atmosphere equations 1.10.

The assumptions taken to estimate Mach Number could introduce errors in CI and TOM predictions. In the study of noise affectation, section 4.5.1., was found the importance of Mach Number. When the Mach Number is affected by noise, the accuracy of the prediction drops off dramatically. In this case, the assumptions of no ISA variation and a calm atmosphere with no wind, affects in the results of the prediction as noise does. Under this circumstances, CI and TOM predictions will be affected, suffering a not quantified variation. Two possible improvements could be applied to avoid this fact:

1. Using a data-based of the real atmosphere in the operating day, to know the real temperature and wind conditions at each point of the trajectory. This data can be take from GRIB formatted files (e.g. those provided from NOAA, National Oceanic Atmospheric Administration, or ECMWF, European Centre for Medium-Range Weather Forecasts).

[4]Horizontal flight which can not be considered a cruise because its duration do not exceed 5 minutes

2. Using ADS-B data instead of DDR2, so this data has been issued by the aircraft and contains all flight performance and on-board parameters due to meteorological conditions.

With this previous assumptions, finally, all the boxes in the table can be filled. And the table required for the input of the algorithm can be introduced in it to start the tests.

4.7.2. Analysis for real routes operated by airlines

Once the table is completed, the analysis will be compose by three studies, one for a low cost airline, another one for a flag carrier and a third one for a one route operted by different airlines. For each one of the airlines will be studied the flights performed for 5 different routes during a period of 6 months. In the following figures is presented an example of a performed test flight.

This first test is focused on route Barcelona-El Prat airport and London Gatwick (LEBL-EGKK). The data selected corresponds to the period from October 2016 to April 2017, with a total number of 472 flights for this route.

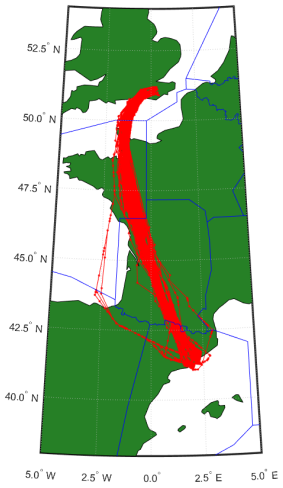


Figure 4.17: Horizontal profile of trajectories LEBL-EGKK

In Figure 4.17 can be seen the horizontal projection of all the trajectories, there is no much dispersion between them, and indicate that this company uses the same waypoints to perform this route. There are a few exceptions that are deviated from the typical route probably to airspace congestions, restrictions or weather issues.

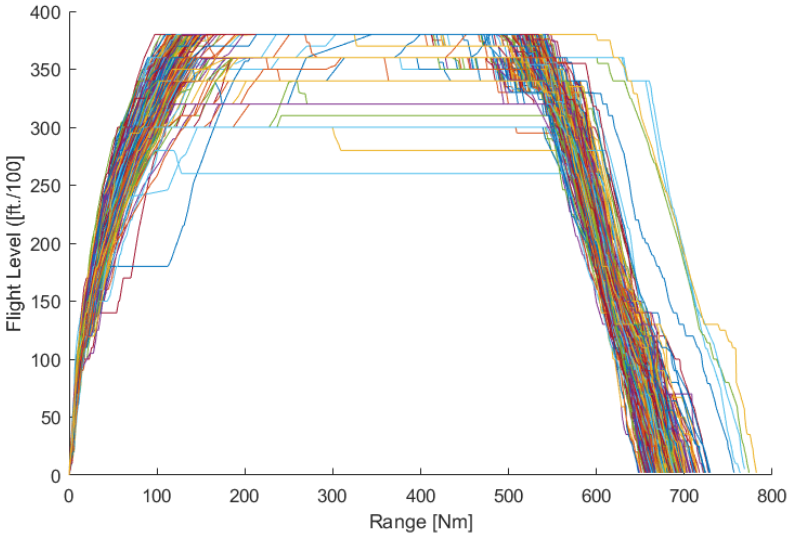


Figure 4.18: Vertical profile of trajectories LEBL-EGKK

In Figure 4.18 can be seen the vertical profile of the route, most flights have covered a range between 650 to 725 Nm, that probably because the execution of different approach procedures, holdings imposed by ATC or has been chosen different waypoints. Also, it is interesting to observe the previously mentioned level-off, that little steps that have to be deleted from the data to obtain the desired trajectory, without ATC affectation. In Figure 4.19 can be seen the representation of a level-off, in blue is represented the desirable trajectory because allows to reach cruise altitude early (FL350), and the orange trajectory has three level-off and implies a later arrival to TOC.

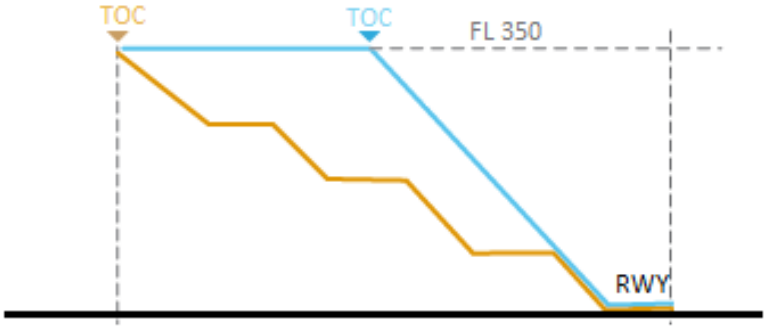


Figure 4.19: Example of level-off during climb

The flight data of the DDR2 platform presents all flights in form of a set of segments where contains times, positions and different flight information. This data permits to make the vertical profile and the route of the flights, but there no information about weather or wind. This means, the algorithms which will be used for ML prediction could not use wind and ISA deviation. To predict CI and TOM, will be used the model showed in Figure 4.14(a) and Figure 4.14(b), where is deleted total time and distance, with an accuracy of 80,2% and 82,3% for the CI and TOM respectively.

To perform the prediction an iterative method will be applied, that is, the predictions of CI and TOM will be do it successively until the mean value of the CI and TOM of iteration N-1 is the same of N. Then the results will be plotted as can be seen in following sections.

4.7.2.1. Airline A - Low cost

This first airline is a low cost airline, and the following routes will be studied:

	Airline A: Low Cost		
	Departure	**Arribal**	**Range [NM]**
Route 1	LEBL	EGKK	683
Route 2	LEBL	EHAM	749
Route 3	LEBL	LIRF	517
Route 4	LEBL	EKCH	1.071
Route 5	LEBL	GCLP	1.238

Table 4.7: Airline A

The data obtained for this 5 routes in the period of 6 months, will be tested with the algorithm of prediction of CI and TOM. Data is represented in a boxplot, separating the results for each different route. The first three route have almost the same range between 500-750 NM while, the fourth and the fifth increment the distance range up to 1.000-1.200 NM.

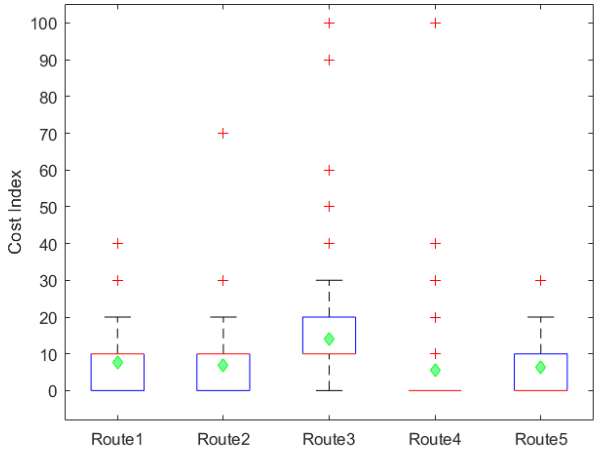


Figure 4.20: Cost Index estimation Airline A.

For the CI values obtained in Figure 4.20, do not exceed the value of 20 in most of the routes, oscillating with very low values of CI. The maximum mean value of CI is found in Route 3, probably for some strategical reason. Also, what can be said is that this low cost airline realizes an strategy of giving more importance to the fuel price than the trip time, that means lower values of CI. In this analysis can be seen that for some routes, like Route 3, the boxplot present the median (red line) on the bottom of the box, that corresponds to the value on the middle of the whole data. Figure 4.21 is the scatter plot for CI in case of Route 3. This plot shows that the great thickness of the predictions are in CI equal to 10, making that the whiskers grow to CI 0 and 30.

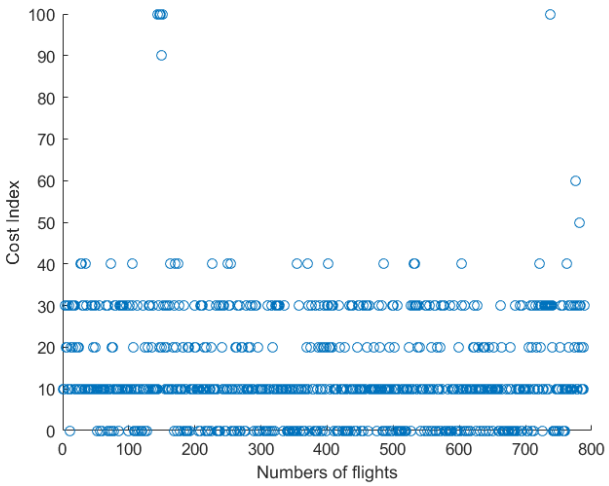


Figure 4.21: CI scatter plot for Route 3

In the case of the TOM study, Figure 4.22, there a significant different between the first three routes and the other two. As the distance range increase and the route is longer, the aircraft will need more fuel to arrive to their destination. Due to an increment of fuel on-board, TOM value is also increased. Route1, 2 and 3 present almost constant range values between 4.700kg and 6.100kg, with a mean value over 5.300kg. In Route4, we can see the predicted results does not present much dispersion. Almost all TOM prediction values are over 61.000-65.000kg. Finally, for Route5 shows more dispersion than Route4 but also have higher TOM values than the first three routes. The augmentation on weight for the last two routes, increment of more than 7.000 kg, could be occasioned by the added fuel. Aircraft will need to fly more distance than routes before, so some fuel must be added to tanks.

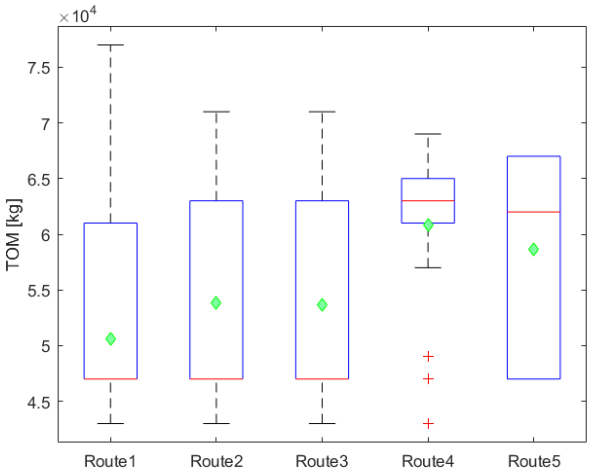


Figure 4.22: Takeoff Mass estimation Airline A.

4.7.2.2. Airline B - Flag carrier

The second study is focused in a flag carrier and we are going to analyzed 5 routes. Those routes are:

	Airline B: Flag Carrier		
	Departure	**Arribal**	**Range [NM]**
Route 1	EGLL	LEBL	651
Route 2	EGLL	LOWW	754
Route 3	EGLL	LPPT	884
Route 4	EGLL	LKPR	611
Route 5	EGLL	LGAV	1.362

Table 4.8: Airline B

As the study on Airline A, data has been obtained for 5 routes in the period of 6 months. Data will be tested the ML algorithm and represent predictions of CI and TOM in a boxplot. Now, the first four routes present similar ranges between 611-884 NM while, the fifth has a range of 1.362 NM.

In the case of a flag carrier CI estimation, the results are shown in Figure 4.23. It shows that the CI used by this flag carrier airline is almost constant. For the studied Routes1-4 of similar ranges, CI value moves between 30 and 40. For the longest route, the fifth one, the mean value of CI decrease to 26. While CI prediction for short ranges are quite precise, values fluctuate between 30 and 40, the longest route present dispersion on their predictions. CI prediction value is below 30, but is not far from the other predicted values. The reason of using higher values of CI is because the operator prioritizes time before fuel consumption. Flag-carrier airlines want to satisfy their passengers to arrive the earlier as possible to their destination.

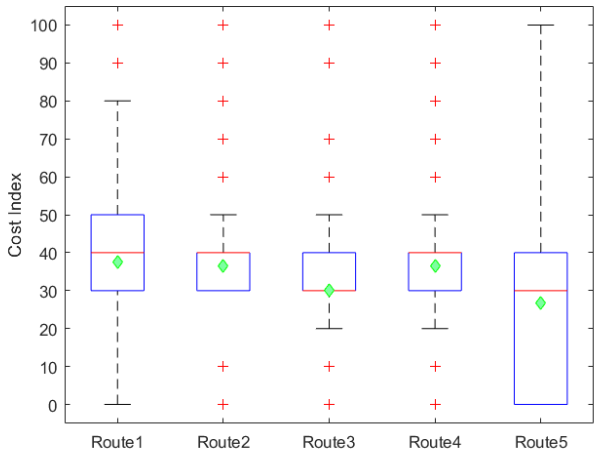


Figure 4.23: Cost Index estimation Airline B.

For TOM predictions, it can be clearly seen the influence of distance. For the lower distance range routes, Route1 and 4, TOM has a high dispersion from lower values of TOM until 6.300kg. Both routes cover over 600 NM and their mean value of TOM also coincides around 5.800kg. Route2 and 3, with similar distance range of 800 NM present TOM around 61.000kg. Finally, Route5 shows higher TOM values than the other routes. Even the distance is notably higher than the previous ones, the increment on TOM is not so evident. TOM acquires a mean value of 63.000kg, but reach TOMs over 71.000kg.

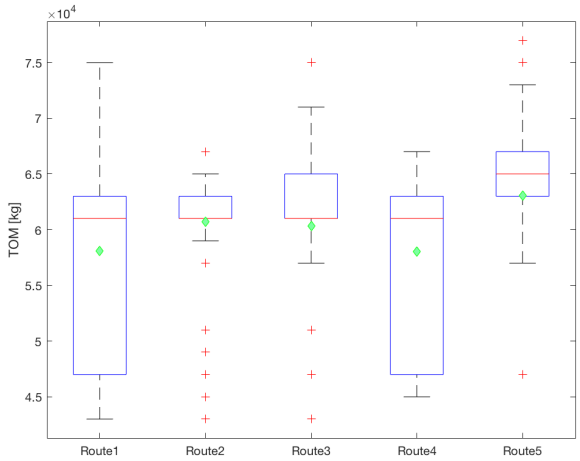


Figure 4.24: Takeoff Mass estimation Airline B.

4.7.2.3. Study of the Route: Barcelona - Dusseldorf

In the following section, is presented a study of a route operated by different airlines. The route has been executed from ”Aeropuerto de Barcelona-El Prat” (LEBL) to ”Dusseldorf International Airport” (EEDL).

The route from LEBL to EEDL is flown everyday by different operators. In this case study, we have only considered the routes operated by an A320. In this conditions, we have found three different airlines which performed this regular flight. Data have been extracted by the DDR2 platform, from the 1 April of 2016 to the 30 April 2017, obtaining a total number of flights of 1.327.

After analyzing each flight, the ML algorithm predict CI and TOM values for each airline. To a better understanding of the information obtained, we have made a boxplot for both CI and TOM prediction, in function of the different airlines. In the boxplot is presented Airline 1 and 3, which are low cost operators, and Airline 2, which corresponds to a flag-carrier operator.

As Figure 4.25 shows, Airline 1 and 3 present very low values of CI prediction. Both mean values correspond to a CI equal to 9, with a dispersion between 0 and 20. This CI results present similarities with the study of Airline A on section 4.7.2.1. As explained before, low-cost operators prioritizes to save fuel cost although they have to spend more time flying. In the boxplot of predicted CI is perfectly both Airlines 1 and 3, uses low values of CI which indicates a predominance of fuel saving. When looking at the Airline 2, CI values slightly increase. Now, CI mean value is 21, with possible results between 10 and 30 but with dispersion on higher CI predictions. Airline 2 is a flag-carrier operator and in contradistinction to low-cost, these airlines do not care about burning much fuel if their passengers could arrive earlier to their destination. They give more importance to time cost than fuel, so CI values accustom to be higher than the previous one. In the boxplot, the CI value appears 12 units above, indicating that the operator gives more importance to time cost than Airline 1 and 3.

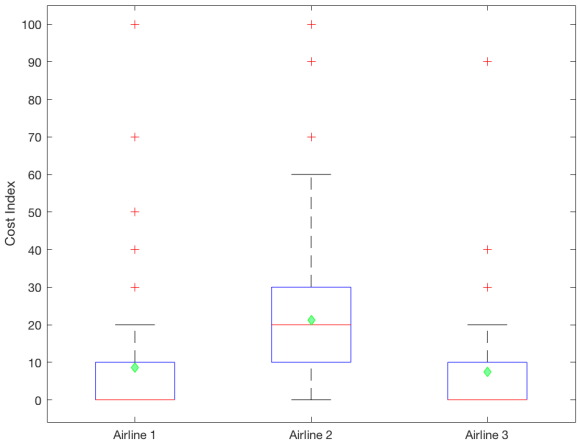


Figure 4.25: Cost Index estimation for route LEBL-EDDL.

For TOM prediction, Figure 4.26, all airlines present similar values. While Airline 1 and 3 show mean values of TOM around 5.700kg, Airline 2 gently increase to 6.000kg. An hypothesis of this raise could be originated by the increment of the on-board fuel. As the second operator presents a higher CI value, the importance of time cost increases. To arrive before to the destination, aircraft velocity must be raised. This change on speed will imply an increment on fuel burn rate. An increment on fuel rate will signify that the aircraft will need to load more fuel on the tank to perform the flight. This increment on fuel could be reflected in the graph as the difference on TOM between Airline 2 and the other two.

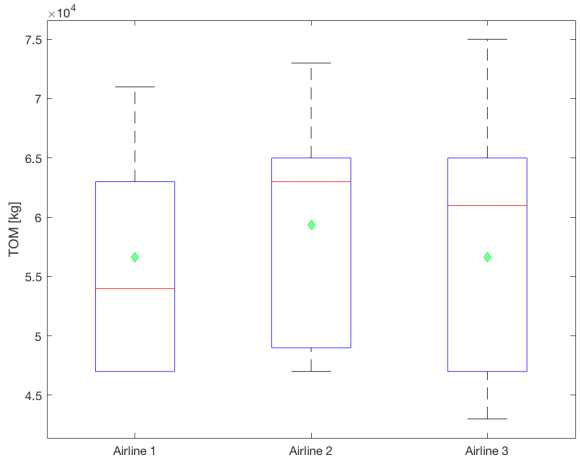


Figure 4.26: Takeoff Mass estimation for route LEBL-EDDL.

4.8. Other computation methods to obtain CI

Finally is interesting to perform a comparison between this method of computing CI that is a Data Based method and will be compared with a Model Based method, which estimats the cost index analytically, based on aircraft performance data and a dynamics model.

Applying Machine Learning is a data base method, because it requires from an input data to be trained and to develop the algorithm to make predictions without a model. Also, this method allows the improvement by adding new data to be trained. On the other hand, Model Based is developed by the application of models and once is developed not requires data to improve.

4.8.1. Model Based

Model Based method, is based on the Method for optimum economy cruise speed in an aircraft [17], but instead of compute the Mach velocity, computes the CI. Their fundamental is similar to the operation of FMS, which receives many inputs, like the weight of the aircraft, the atmospheric pressure and temperature, the wind speed and it is expected to receive the CI value of the flight to compute the optimal Mach. However, in this case the Mach is rather the cost index, and it is supposed to be the optimal one.

The FMS works by reducing the variable costs, that can be defined as seen in equation 1.8. The equation can be described as a function of mass, wind, CI. To find the optimal for the equation of total cost of the trip, it is derived from the Mach, and equated to zero, so the optimum mach is obtained as equation 4.4.

$$\frac{\partial f}{\partial M}[Mass, CI, Wind, \Delta T] \rightarrow M_{Optimum}(Mass, CI, Wind) \tag{4.4}$$

The Model Based method presented, uses the same expression as FMS but finding the optimal CI, in function of the Mach Number. From equation above, CI is obtained as a function of mass, optimum mach, wind and temperature, because other data is an input for the calculus.

This method only requires data from two points of the cruise phase and computes the CI as the average of the CI of boh points, while the Data Based method requires of the all trajectory. The drawback of the model-based method is that it requires for the weight value of the airplane in the point, and for example in a same FL of cruise the mass of the aircraft decreases in as fuel is burned. On the other method, the data base, weight is not required, because CI is computed at the same time as TOM, with an iterative method.

CONCLUSIONS

CI and TOM are two parameters that notably affects on the aircraft trajectory and describes the airlines operative strategy. For this reason, the trajectory prediction will never correspond exactly to the real life operative because of the non-availability of these parameters.

In this project has been developed a new technique to estimate the CI and TOM applying Machine Learning. From a FMS simulator called PEP from Airbus has been generated a database of trajectories. With this database, Machine Learning creates a model based on the patterns for CI and TOM based on the trajectories. This pattern is embodied in an algorithm which allows the estimation of CI and TOM for a new performed trajectory.

The best way to face a problem of trajectory resolution is using Ensembled Bagged Trees model. A bagged decision tree model consists of trees that are trained independently on data that is bootstrapped from the input data. This algorithm has been selected not only because it has presented the best accuracy results, rather it allow flexibility on their overfitting situations by changing the configuration parameters involved in their complexity. Once selected the algorithm, a study of the input parameters have to be done in order to improve the estimations. Improve the estimations not only implies to know which are the main relevant variables, also the robustness of the algorithm in front of the noise affectations.

In the case of CI, the most relevant input variable is the Mach Number because it is the most visible evidence given to the time-fuel cost relation. A slight variation of 8% on the Mach velocity, could drop off the accuracy of the model in more than a 60%. The flight phases more affected by the CI chosen are the climb and the descent, particularly on the climb or descent slope to TOC or from TOD, respectively. While CI is strongly related with Mach velocity, TOM is more related with the cruise altitudes (FL) or flight distances. Also TOC has an influence on TOM estimation because weight limits the TOC altitude due to lift and rate of climb restrictions. TOM estimation presents a higher robustness in front noise than CI. But when the algorithm validation is performed, it is verified that original accuracy from TOM estimation falls due to an overfitting between the trained and the test data.

The final goal of this algorithm is to be used for real application cases. The handicap of this study is that the results will never be validated due to the privacy of CI and TOM values imposed by the airlines. What can be said about this study is that CI value will be underestimated due to the assumptions made to obtain the Mach Number. This Mach Velocity is obtained from ISA and a calm atmosphere.In spite of the values of CI and TOM would not correspond to the real ones, a pattern strategy by the operators have been observed. The low cost airlines based its operations on fuel saving which implies lower CI, while flag-carriers prefer to minimize trip time, using higher values of CI. Also, for operations with higher CI, has been found a increment of TOM because of the increase on fuel flow.

This method could be characterized as a Data Based method. Nowadays, the estimation of CI and TOM is gaining relevance because allows better trajectory prediction, useful in programs in development such us SESAR and NextGEN. That interest has made that new methods emerged like Model Based, but unlike Data Based requires aircraft weight as an input.

BIBLIOGRAPHY

[1] Flight Operations Bill Roberson, Senior Safety Pilot. Cost Index Explained. (Ci):26–28. ix, 7, 8

[2] E Ulfbratt and J McConville. Comparison of the SESAR and NextGen Concepts of Operations. *NCOIC Aviation IPT*, 1.0:22, 2008. 1

[3] Richard Alligier, David Gianazza, Nicolas Durand, R Alligier, D Gianazza, and N Durand. Learning To cite this version : Predicting Aircraft Descent Length with Machine Learning. 2016. 1

[4] R Alligier, D Gianazza, and N Durand. Machine Learning Applied to Airspeed Prediction During Climb. In *11th USA/Europe Air Traffic Management Research and Development Seminar*, pages 1–10, 2015. 1

[5] Eurocontrol. Continuous Climb and Descent Operations. 4

[6] Customer Services AIRBUS. Cost Index. (II):1–104, 1998. 5

[7] Customer Services AIRBUS. Aircraft performance. 10

[8] Agnieszka Ławrynowicz and Volker Tresp. Introducing Machine Learning. *Perspectives on Ontology Learning*, 2014. 12, 19

[9] Eurocontrol. DDR2 Quick Reference Guide. *Change*. 12

[10] MATLAB machine learning with matlab. https://es.mathworks.com/solutions/machine-learning.html. Accessed: 2017-04-21. 19

[11] Trevor Hastie, Robert Tibshirani, and J. H. (Jerome H.) Friedman. *The elements of statistical learning : data mining, inference, and prediction*. Springer, New York [etc.] :, 2nd ed. edition, 2009. 19

[12] Nello Cristianni and John Shawe-Taylor. CAn Introduction to Support Vector Machines and Other Kernel-based Learning Methods., 2000. 20

[13] Leo Breiman, Jerome H. Friedman, Richard A. Olshen and Charles J. Stone. Classification and Regression Trees., 1984. 21

[14] Microsoft how to choose algorithms for microsoft azure machine learning. https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice. Accessed: 2017-03-10. 22

[15] I. T. Jolliffe. *Principal component analysis*. Springer, New York [etc.] :, 2nd ed. edition, 2002. 27

[16] Customer Services AIRBUS. A320 Family performance retention and fuel savings. *Fuel*, 2008. 29

[17] Eduardo Gallo Olalla. Method for optimum economy cruise speed in an aircraft.pdf, 2016. 59

[18] AIRBUS aircraft characteristics. http://www.airbus.com/aircraftfamilies/passengeraircraft/aircraft-characteristics. Accessed: 2017-02-16. 67

APÈNDIXS

APPENDIX A. CHARACTERISTICS WEIGHTS FOR AIRBUS

Table A.1 shows the characteristics weights for Airbus aircraft models [18].

AIRCRAFTS	MTOW	MLW	MZFW	OEW	MFL	MPL
A318-112	68000	57500	54500	40900	18800	13600
A319-114	70000	61000	57000	39000	18394	18000
A320-212	77000	64500	60500	39000	18730	21500
A321-111	83000	73500	69500	47000	24044	22500
A330-243	230000	180000	168000	124948	109555	43052
A330-342	212000	174000	164000	126000	77060	38000
A340-212	257000	181000	169000	126000	77060	38000
A340-313	257000	186000	174000	131000	108831	43000
A340-541	373200	243000	230000	182800	174400	47200
A340-642	380000	265000	251000	187000	153800	64000
A380-841	569000	391000	255000	282500	254781	83500

Table A.1: Characteristics weights for Airbus

APPENDIX B. PYTHOON LOOPS FOR AIRBUS AIRCRAFT MODELS

Table B.1 shows the computations for the inputs used to obtain the trajectories in the python code. The number of loops is computed as:

$$Interactions = \sum_{short}^{long} \prod \frac{Property_{max} - Property_{min}}{\Delta Property} \tag{B.1}$$

	Long Routes			Short Routes		
Properties	Minimum	Maximum	Increment	Minimum	Maximum	Increment
MTOW	47000	77000	2000	43000	67000	2000
CI	0	100	10	0	100	10
RANGE	400	3300	270	150	300	150
WIND	-80	80	20	-80	80	20
ISA	-20	20	10	-20	20	10
ALTITUDE	26000	27000	1000	19000	20000	1000
	Total iterations			**111.100**		

Table B.1: Airbus A320-212

APPENDIX C. INPUT AND OUTPUT MODEL FROM PEP

This is an example of an Input followed by an Output:

```
     Input file and databases file specifications:
     ---------------------------------------------
     Input File=   c:\users\icarus\desktop\aircrafts\a320-212\51000_0
     Aero databas  C:\Program Files\PEP\Data\A32021FM.BDC
     Gene databas  C:\Program Files\PEP\Data\G32021FM.BDC
     Moto databas  C:\Program Files\PEP\Data\M565A1FM.BDC

01.BLOCK : GENERAL INPUT ================================================
DATE   =4MAY02
DT     =      010 CG     =    27.00 RCABDE =     350. FUELM  =   18730.
WMTOW  =   77000. WMLAN  =   64500. WMOE   =   39000. WMZF   =   60500.
FACFU  =    1.000 FACTR  =    1.000 FACBUF =    1.300 FACCW  =    1.000
FLHV   =   18590.
02.BLOCK : CONTROL STATEMENTS ===========================================
KDIM   = 3        KMIOUT = 1        KWGHT  = 1
KAC    = 2        KAI    = 0        KENG   = 0        KLKED  = 0
KACEF  = 1        KAICP  = 0        KGEAR  = 0        KFTP   = 0
KMRTT  = 0
03.BLOCK : TAKE OFF + INITIAL CLIMB =====================================
FTAKO  =       .0 DTAKO  =       .0 TTAKO  =       .0
ATAKO  =       .0 FTAXOF =    140.0 TTAXOF =     12.0
KTAKO  = 1
04.BLOCK : CLIMB ========================================================
CASCL1 =   250.00 CASCL2 =   300.00 XMCL   =     .780 ACLIMS =   10000.
RCLIMB =     300.
05.BLOCK : CRUISE =======================================================
VCRUI  =     .780 ASTEP1 =    2000. TIMECR =       5. DELTAV =     .000
DSTEP1 =        . DSTEP2 =        . DSTEP3 =        .
KLORA  = 0        KRAT   = 1        KSTEP  = 6
ASTEP2 =    2000. ASTEP3 =    2000. ASTEP4 =    2000.
ASTEP5 =    2000. ASTEP6 =    2000.        .
DSTEP4 =        . DSTEP5 =        . DSTEP6 =        .
06.BLOCK : DESCENT ======================================================
CASDE1 =   300.00 CASDE2 =   250.00 XMDE   =     .780 ADESCS =   10000.
KAB    = 0
09.BLOCK : APPROACH AND LANDING =========================================
FGOARO =        . FMISAP =        .
TGOARO =        . TMISAP =        .
ALAND  =        . FLAND  =        . DLAND  =        . TLAND  =        .
KLAND  = 1        KIFR   = 1
10.BLOCK : RESERVES =====================================================
KRESER = 1        KRESA1 = 1        KRESA2 = 1        KRESE3 = 1
FACRES =    1.050 FACRA1 =    1.000 FACRA2 =    1.050 FACRE3 =    1.030
FADRES =        . FADRA1 =        .
12.BLOCK : SPECIAL INPUTS ===============================================
WINDDP =      .00 WINDCR =      080 WINDLA =      .00 WINFIX =      .00
KCONTA = 0        WINCHG =      .00
R1     =     2430 R2     =        . DR     =        .
NW     = 1
XW1    =   051000 XW2    =   55556. XW3    =   55557. XW4    =   55558.
NA     = 1
XA1    =    26000 XA2    =   33334. XA3    =   33335. XA4    =   33336.
13.BLOCK : TEXT INPUT ===================================================
LTEXT  =A320    STANDARD FLIGHT PLANNING
LDP    =XXXX      LDS    =YYYY      LAL    =ZZZZ
17.BLOCK : FMS CONTROL STATEMENTS =======================================
KFMSCL = 1        KFMSCR = 1        KFMSDE = 1        KFMSHO = 1
CINDEX =      010 KDIMCX = 1
99----==> end of user input file ========================================


PROGRAM : FLIP25F2    23.01.06

AERO    : A320-111/211/212    05/10/92
```

ENGINE : A320-111/211/212 05/10/92

GENERAL : A320-111/-210 14/06/96

A320 DATABASE WITH ENGINE CFM56/5-A

1. PICTURE

A320 STANDARD FLIGHT PLANNING

F L I G H T P L A N N I N G

FMS - CALCULATION (HONEYWELL)

FROM :XXXX TO :YYYY

AIRCRAFT : A320-111/211/212 ENGINE : CFM56/5-A DATE : 4MAY02

AIR CONDITIONING : LOW ANTI ICING : OFF FUEL HEATING VALUE : 18590 BTU/LB

TEMPERATURE : ISA +10 CG POSITION : 27.0 % MAX CABIN RATE OF DESCENT : 350 FT/MIN

CRUISE ALTITUDE : 38000 FT TAKEOFF WEIGHT : 51000 KG OPERATING WEIGHT EMPTY : 39000 KG

COST INDEX : 10 (KG/MIN)

CLIMB-MODE : ECONOMIC CRUISE-MODE : ECONOMIC DESCENT-MODE : ECONOMIC

INSTALLED FACTORS : RESERVES 5.0 % (on fuel)

POINT OF FLIGHT	WEIGHT	FUEL	TIME	DISTANCE		FL	SPEEDS	WIND	OAT
				GRND	AIR				
	KG	KG	MIN	NM		--	KT / M	KT	CELSIUS
RAMP WEIGHT	51140								
TAXI OUT		140	12			0			25.0
WEIGHT AT BRAKE RELEASE	51000							0	
TAKEOFF and INITIAL CLIMB		140	1	3	3	15		5	
CLIMB		698	8	51	45		250/297/.601		
CLIMB		163	3	21	17		.625		
CLIMB		168	3	23	19		.648		
CLIMB		173	3	26	21		.672		
CLIMB		182	4	30	24		.696		
CLIMB		198	4	36	29		.719		
CLIMB		117	3	27	21		.725		
CRUISE (7th flight level)		6662	229	2069	1623	380	.725 ECONOMIC	117	-46.5
DESCENT to DESTINATION AIRPORT		160	20	131	111	15	.709/250/250	3	
IFR APPROACH and LANDING		108	6	13	13	0		0	25.0
TRIP FUEL		8769	284	2430	1926				
ROUTE RESERVES (5.0% trip fuel)		438							
WEIGHT AT DESTINATION AIRPORT	42231								
BLOCK FUEL		9347	296						

DISTANCE from DEPARTURE AIRPORT to DESTINATION AIRPORT 2430 / 1926 NM (GROUND / AIR)

RAMP WEIGHT	51140 KG	TOTAL FUEL on BOARD	9347 KG	TOTAL TIME	4/56 H/MIN
TAKEOFF WEIGHT	51000 KG	TRIP FUEL	8769 KG	TRIP TIME	4/44 H/MIN
LANDING WEIGHT	42231 KG	ROUTE RESERVES	438 KG		
ZERO FUEL WEIGHT	41793 KG				
PAYLOAD	2793 KG				

APPENDIX D. MACHINE LEARNING TRAINING RESULTS

In this Appendix are shown the results obtained by applying the available Machine Learning methods in Matlab. Figure D.1 represents the training made for TOM predictions, while the second one, Figure D.2, represents the training for CI predictions.

(a) Decision trees (22 variables)

(b) Decision trees (23 variables)

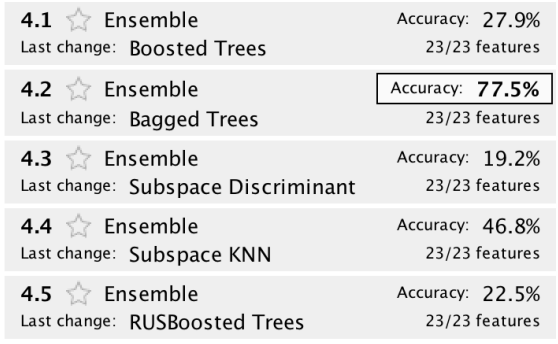
(c) Discriminant analysis (22 variables)

(d) Discriminant analysis (23 variables)

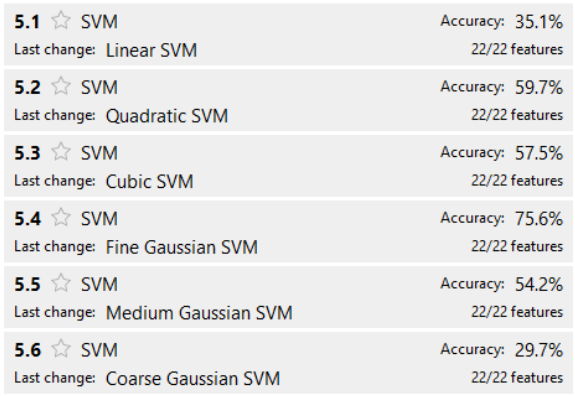
(e) KNN (22 variables)

(f) KNN (23 variables)

(g) Ensemble Classifier (22 variables)



(h) Ensemble Classifier (23 variables)



(i) SVM (22 variables)

(j) SVM (23 variables)

Figure D.1: TOM Training for A-320

1.1 ☆ Tree	Accuracy: 49.4%	
Last change: Complex Tree	22/22 features	
1.2 ☆ Tree	Accuracy: 38.2%	
Last change: Medium Tree	22/22 features	
1.3 ☆ Tree	Accuracy: 29.7%	
Last change: Simple Tree	22/22 features	

(a) Decision trees (22 variables)

1.1 ☆ Tree	Accuracy: 51.0%	
Last change: Complex Tree	23/23 features	
1.2 ☆ Tree	Accuracy: 37.7%	
Last change: Medium Tree	23/23 features	
1.3 ☆ Tree	Accuracy: 29.6%	
Last change: Simple Tree	23/23 features	

(b) Decision trees (23 variables)

2.1 ☆ Linear Discriminant	Accuracy: 20.5%	
Last change: Linear Discriminant	22/22 features	
2.2 ☆ Quadratic Discriminant	Accuracy: 16.6%	
Last change: Quadratic Discriminant	22/22 features	

(c) Discriminant analysis (22 variables)

2.1 ☆ Linear Discriminant	Accuracy: 20.1%	
Last change: Linear Discriminant	23/23 features	
2.2 ☆ Quadratic Discriminant	Accuracy: 16.3%	
Last change: Quadratic Discriminant	23/23 features	

(d) Discriminant analysis (23 variables)

3.1 ☆ KNN	Accuracy: 41.9%	
Last change: Fine KNN	22/22 features	
3.2 ☆ KNN	Accuracy: 39.8%	
Last change: Medium KNN	22/22 features	
3.3 ☆ KNN	Accuracy: 39.3%	
Last change: Coarse KNN	22/22 features	
3.4 ☆ KNN	Accuracy: 41.9%	
Last change: Cosine KNN	22/22 features	
3.5 ☆ KNN	Accuracy: 38.3%	
Last change: Cubic KNN	22/22 features	
3.6 ☆ KNN	Accuracy: 41.4%	
Last change: Weighted KNN	22/22 features	

(e) KNN (22 variables)

3.1 ☆ KNN	Accuracy: 37.8%	
Last change: Fine KNN	23/23 features	
3.2 ☆ KNN	Accuracy: 37.1%	
Last change: Medium KNN	23/23 features	
3.3 ☆ KNN	Accuracy: 38.4%	
Last change: Coarse KNN	23/23 features	
3.4 ☆ KNN	Accuracy: 39.1%	
Last change: Cosine KNN	23/23 features	
3.5 ☆ KNN	Accuracy: 36.2%	
Last change: Cubic KNN	23/23 features	
3.6 ☆ KNN	Accuracy: 37.7%	
Last change: Weighted KNN	23/23 features	

(f) KNN (23 variables)

4.1 ☆ Ensemble	Accuracy: 41.1%	
Last change: Boosted Trees	22/22 features	
4.2 ☆ Ensemble	Accuracy: **76.7%**	
Last change: Bagged Trees	22/22 features	
4.3 ☆ Ensemble	Accuracy: 34.5%	
Last change: Subspace Discriminant	22/22 features	
4.4 ☆ Ensemble	Accuracy: 25.1%	
Last change: Subspace KNN	22/22 features	
4.5 ☆ Ensemble	Accuracy: 39.6%	
Last change: RUSBoosted Trees	22/22 features	

(g) Ensemble Classifier (22 variables)

4.1 ☆ Ensemble	Accuracy: 41.8%	
Last change: Boosted Trees	23/23 features	
4.2 ☆ Ensemble	Accuracy: 77.1%	
Last change: Bagged Trees	23/23 features	
4.3 ☆ Ensemble	Accuracy: 36.7%	
Last change: Subspace Discriminant	23/23 features	
4.4 ☆ Ensemble	Accuracy: 18.6%	
Last change: Subspace KNN	23/23 features	
4.5 ☆ Ensemble	Accuracy: 40.0%	
Last change: RUSBoosted Trees	23/23 features	

(h) Ensemble Classifier (23 variables)

5.1 ☆ SVM	Accuracy: 54.2%	
Last change: Linear SVM	22/22 features	
5.2 ☆ SVM	Accuracy: 75.4%	
Last change: Quadratic SVM	22/22 features	
5.3 ☆ SVM	Accuracy: 49.0%	
Last change: Cubic SVM	22/22 features	
5.4 ☆ SVM	Accuracy: 67.5%	
Last change: Fine Gaussian SVM	22/22 features	
5.5 ☆ SVM	Accuracy: 67.5%	
Last change: Medium Gaussian SVM	22/22 features	
5.6 ☆ SVM	Accuracy: 51.0%	
Last change: Coarse Gaussian SVM	22/22 features	

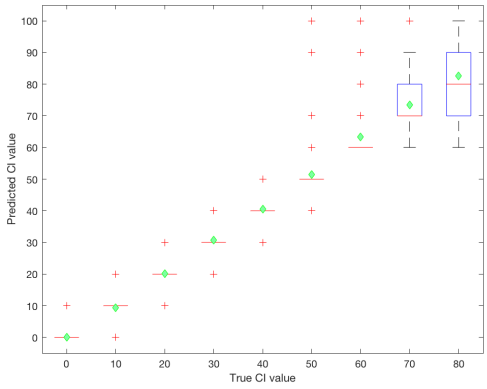
(i) SVM (22 variables)

5.1 ☆ SVM	Accuracy: 56.3%	
Last change: Linear SVM	23/23 features	
5.2 ☆ SVM	Accuracy: **77.4%**	
Last change: Quadratic SVM	23/23 features	
5.3 ☆ SVM	Accuracy: 69.7%	
Last change: Cubic SVM	23/23 features	
5.4 ☆ SVM	Accuracy: 69.2%	
Last change: Fine Gaussian SVM	23/23 features	
5.5 ☆ SVM	Accuracy: 71.1%	
Last change: Medium Gaussian SVM	23/23 features	
5.6 ☆ SVM	Accuracy: 52.8%	
Last change: Coarse Gaussian SVM	23/23 features	

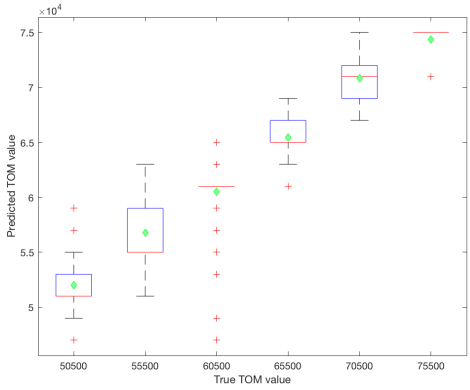
(j) SVM (23 variables)

Figure D.2: CI Training for A-320

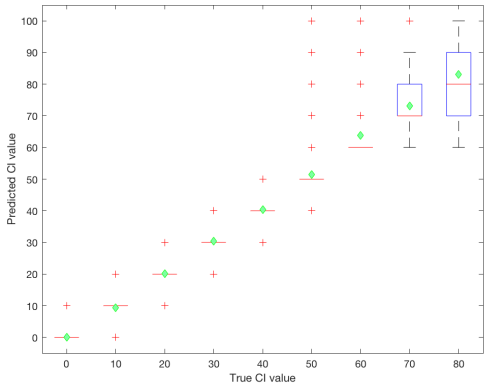
APPENDIX E. OVERFITTING STUDY FOR PEP DATA TEST



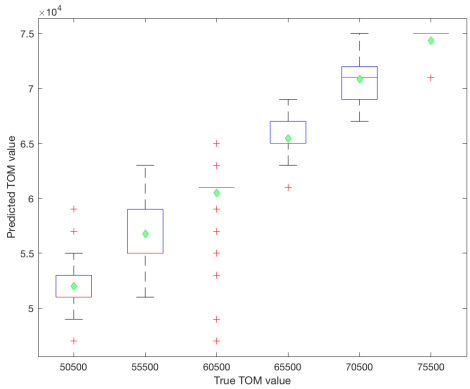
(a) CI: Number of splits = 5



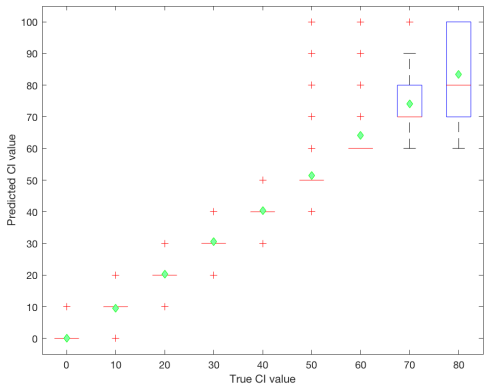
(b) TOM: Number of splits = 5



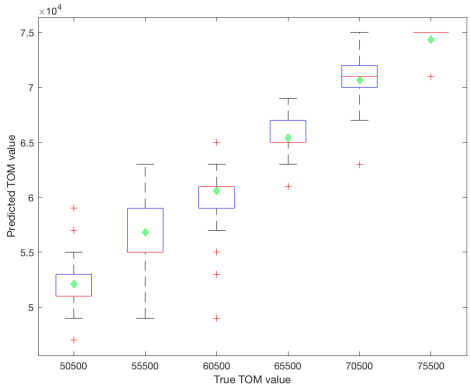
(c) CI: Number of splits = 15



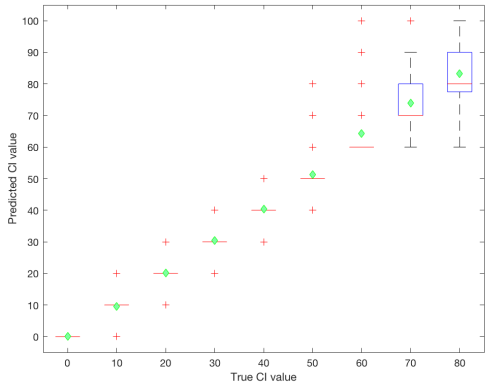
(d) TOM: Number of splits = 15



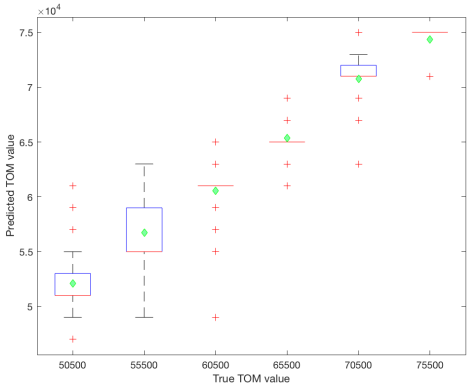
(e) CI: Number of splits = 20



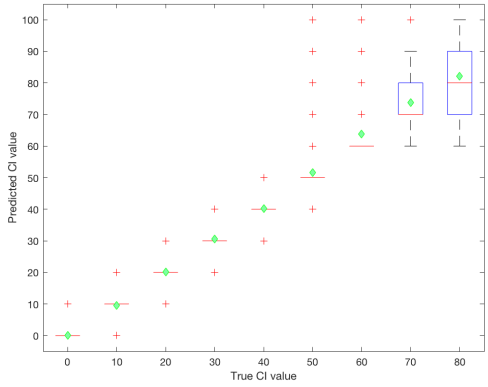
(f) TOM: Number of splits = 20



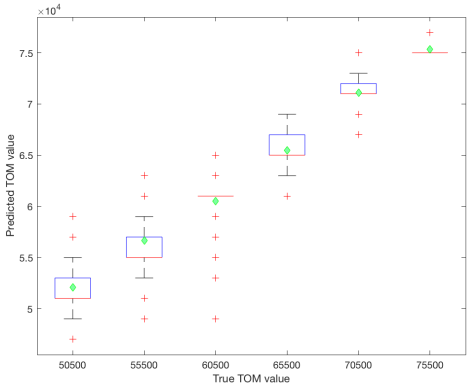
(g) CI: Number of splits = 30



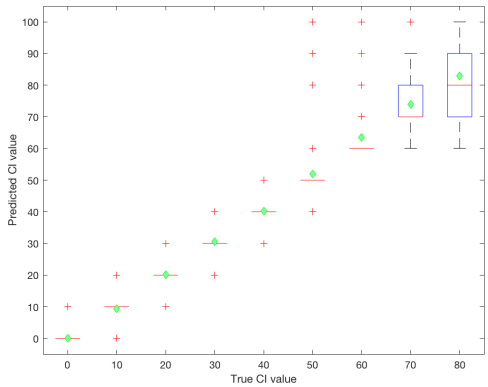
(h) TOM: Number of splits = 30



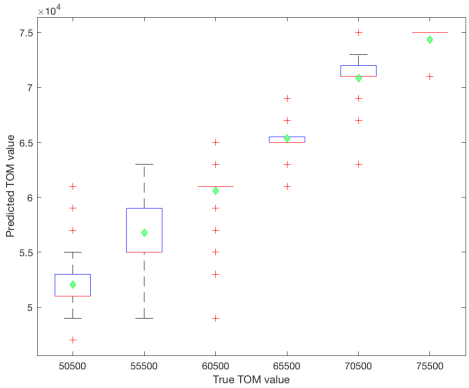
(i) CI: Number of splits = 35



(j) TOM: Number of splits = 35

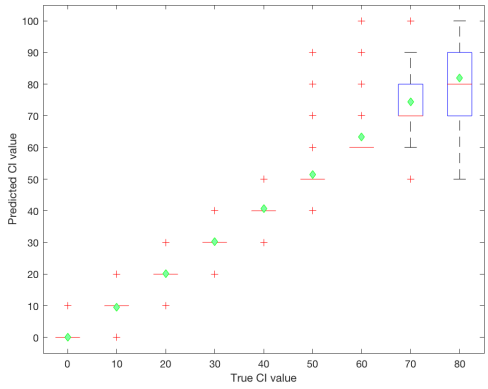


(k) CI: Number of splits = 50

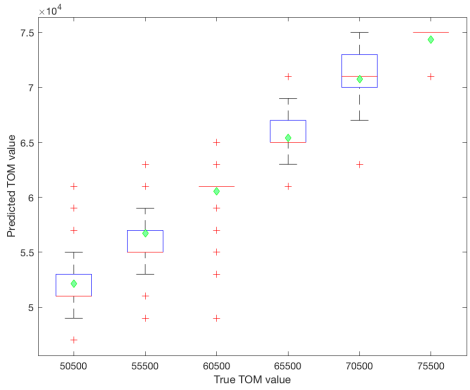


(l) TOM: Number of splits = 50

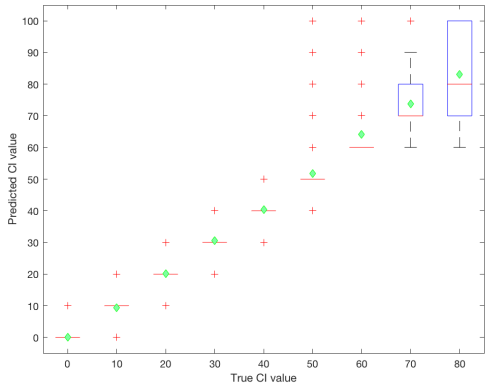
Figure E.1: PEP validation changing number of splits



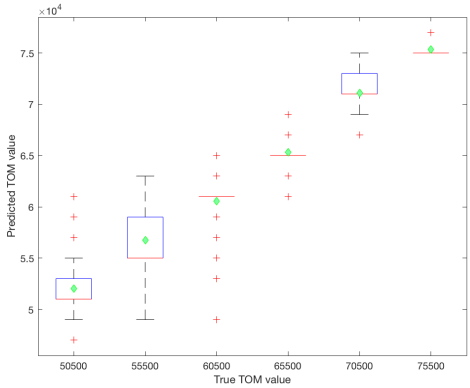
(a) CI: Number of learners = 50



(b) TOM: Number of learners = 50



(c) CI: Number of learners = 200



(d) TOM: Number of learners = 200

Figure E.2: PEP validation changing number of learners

	Percentage of hits	
Number of splits	**TOW**	**CI**
5	68,49%	81,51%
15	68.48%	81,05%
20	67,77%	81,05%
30	68,23%	82,23%
35	68,69%	82,02%
50	68,55%	81,64%

Table E.1: Hits results for changing the number of splits

	Percentage of hits	
Number of learners	**TOW**	**CI**
50	57,71%	81,38%
200	67,77%	80,99%

Table E.2: Hits results for changing the number of learners