



**TecnoCampus**  
Escola Superior  
Politécnica

Centre adscrit a la



Universitat  
Pompeu Fabra  
Barcelona

**Enginyeria Mecànica**

**MANTENIMENT PREDICTIU D'UN VENTILADOR AMB  
INTEL·LIGÈNCIA ARTIFICIAL**

**Memòria**

**PAU BARRULL AYMAT  
PONENT: CARLES PAUL RECARENS**

**TARDOR/PRIMAVERA 2023**



**TecnoCampus**  
Mataró-Maresme

## **Resum**

L'objectiu d'aquest TFG és investigar la possibilitat d'implementar un sistema de manteniment predictiu fonamentat en l'ús d'intel·ligència artificial. El treball pretén implementar el manteniment predictiu en un ventilador TROTEC TVE 14, de manera que, mesurant les vibracions/acceleracions del ventilador amb un ordinador Raspberry Pi 4 i un acceleròmetre Adafruit ADXL345 es puguin predir possibles errors. La finalitat d'aquesta idea és trobar una solució de baix cost que allargui l'esperança de vida del ventilador i redueixi els costos de manteniment. Gràcies a les vibracions detectades per l'acceleròmetre, el model d'intel·ligència artificial desenvolupat podrà detectar si el ventilador està funcionant amb normalitat. Aquest estudi estableix una base per desenvolupar un sistema basat en tecnologia de baix cost i intel·ligència artificial amb el que es podrien predir avaries amb precisió, proporcionant una solució fiable i econòmica per al manteniment del ventilador o altra maquinària industrial del món real.

## **Agraïments**

En aquest últim pas per a obtenir el títol com a enginyer mecànic, m'agradaria agrair a tots els que m'han mostrat suport en tot el procés. Ha sigut un camí ple d'alts i baixos que semblava que mai arribava a la seva fi, però finalment aquí estem. Voldria agrair sobretot al meu entorn més pròxim, familiars i amics, per tot el suport mostrat en aquests anys i per empènyer a superar-me dia rere dia. També m'agradaria agrair al meu tutor Carles Paul per guiar-me i sobretot fer-me qüestionar a mi mateix com fer les coses en aquest treball. Vull agrair especialment a la meva parella, que és la que ha hagut de suportar-me més durant aquest any, ja que ha sigut un procés ple de diferents emocions com la ràbia, la frustració, l'alegria, l'adrenalina, la tristesa i finalment una sensació de pau i plenitud amb l'entrega d'aquest treball. Finalment, vull aprofitar aquest apartat per donar tot el meu suport a la gent que com jo, treballa i estudia a la vegada. Nosaltres sabem de sobre que el sistema educatiu no està dissenyat ni s'adapta a nosaltres, però també s'ha de reconèixer que tot i que és una etapa extremadament dura també ho és en proporció gratificant. A tota aquesta gent, molta merda!

# Índex.

1. Glossari de termes.....	VII
2. Introducció .....	9
2.1 Context i motivació .....	9
2.2 Objectius i abast.....	9
2.3 Estructura del TFG .....	10
3. Fonaments teòrics .....	11
3.1 Manteniment.....	11
3.1.1 Tipus de manteniment .....	11
3.1.2 Beneficis i aplicacions del manteniment predictiu en la indústria.....	12
3.1.3 Tècniques i eines utilitzades en el manteniment predictiu .....	13
3.2 Intel·ligència artificial: .....	14
3.2.1 Concepte d'intel·ligència artificial.....	14
3.2.1.1 Aprenentatge supervisat, no supervisat i per reforç .....	15
3.2.1.2 Aprenentatge automàtic (machine learning) vs. aprenentatge profund (deep learning).....	16
3.3 Algorismes, xarxes i funcions per a l'aprenentatge automàtic i l'aprenentatge profund: .....	17
3.3.1 Algorismes d'aprenentatge automàtic:.....	17
3.3.2 Xarxes neuronals i arquitectures d'aprenentatge profund.....	18
3.3.3 Funcions d'activació, optimització, funció de pèrdua i regularització:.....	18
3.4 Processament de senyals i extracció de característiques .....	19
3.4.1 Conceptes bàsics de processament de senyals:.....	20
3.4.2 Extracció de característiques:.....	20
3.5 Raspberry Pi i acceleròmetres .....	21
3.5.1 Raspberry Pi 4: característiques, especificacions i funcionalitats .....	21
3.5.2 Acceleròmetres: funcionament, tipus i aplicacions, amb èmfasis en el Adafruit ADXL345 .....	22
3.5.3 Protocol de comunicació I2C .....	23
3.6 Llibreries i eines de programació .....	24
3.6.1 Python .....	24
3.6.2 TensorFlow .....	25
3.6.3 Scikit-learn .....	25
3.6.4 Edge Impulse.....	25
3.7 Validació i avaluació de models d'intel·ligència artificial.....	26
3.7.1 Mètriques de rendiment.....	26

4.	Implementació del sistema d'adquisició de dades .....	30
4.1	Descripció general del sistema d'adquisició de dades .....	30
4.2	Connexions físiques entre la Raspberry Pi 4 i l'acceleròmetre Adafruit ADXL34530	
4.3	Posicionament del sensor per obtenir les vibracions òptimes .....	32
4.4	Càlcul de la freqüència de mostreig .....	34
4.4.1	Mesurament de les revolucions per minut del ventilador amb un tacòmetre làser .....	34
4.4.2	Teorema de mostreig de Nyquist-Shannon .....	37
5.	Model d'AA amb Scikit-learn .....	39
5.1	Adquisició i emmagatzematge de dades de l'acceleròmetre pel model d'AA.....	39
5.1.1	Inicialització i configuració de l'acceleròmetre .....	39
5.1.2	Descripció del codi de recollida de dades pel model d'AA.....	39
5.2	Preprocessament de les dades del model d'AA.....	41
5.2.1	Descripció del codi pel preprocessament de les dades del model d'AA .....	41
5.3	Entrenament i avaluació del model d'AA .....	43
5.3.1	Selecció i justificació de l'algorisme d'entrenament del model d'AA.....	43
5.3.2	Descripció del codi per l'entrenament del model d'AA.....	45
5.3.3	Avaluació del model d'AA: .....	48
5.4	Desplegament del model d'aprenentatge automàtic .....	55
5.4.1	Descripció del codi per al desplegament del model d'aprenentatge automàtic .....	55
6.	Model d'AP amb Edge Impulse.....	59
6.1	Adquisició i emmagatzematge de dades per el model d'AP amb Edge Impulse .	59
6.1.1	Connexió de la Raspberry Pi a la plataforma Edge Impulse .....	59
6.1.2	Descripció del codi de recollida de dades per el model d'AP amb Edge Impulse.....	63
6.1.3	Separació de les dades en grups d'entrenament i prova .....	66
6.2	Disseny general del model d'AP amb Edge Impulse.....	68
6.3	Preprocessament i caracterització espectral de les dades amb Edge Impulse .....	71
6.3.1	Filtratge de les Dades amb Edge Impulse .....	71
6.3.2	Anàlisi Espectral de les dades amb Edge Impulse .....	72
6.3.3	Gràfic de dispersió de les dades preprocessades .....	74
6.4	Entrenament del model d'AP amb Edge Impulse .....	75
6.5	Validació del model d'AP amb Edge Impulse .....	78
6.5.1	Exactitud i Pèrdua del Model .....	79
6.5.2	Anàlisi de la Matriu de Confusió .....	79
6.5.3	Puntuació F1.....	80

6.5.4	Gràfica de dispersió posterior a l'entrenament de les dades .....	80
6.6	Introducció d'una anomalia al model d'AP amb Edge Impulse.....	81
6.6.1	Validació del model d'AP amb Edge Impulse introduint l'anomalia .....	84
6.7	Testing del model d'aprenentatge profund amb Edge Impulse. ....	85
7.	Resultats i discussió .....	87
8.	Conclusions i treballs futurs.....	88
8.1	Conclusions .....	88
8.2	Limitacions i treballs futurs .....	91
9.	Referències .....	93

## Índex de figures.

<b>Figura - 1</b>	Ubicació pins de connexió de la Raspberry Pi 4. Imatge obtinguda de la referència [6].....	31
<b>Figura - 2</b>	Esquema elèctric de les connexions entre l'ADXL345 i la Raspberry Pi 4 (imatges realitzades amb el programa Fritzing).....	31
<b>Figura - 3</b>	Comparació exactitud, posició 1 vs. posició 2.....	33
<b>Figura - 4</b>	Possibles posicionaments del sensor en el ventilador.....	34
<b>Figura - 5</b>	Cartó negre amb cinta reflectant del tacòmetre .....	35
<b>Figura - 6</b>	Tacòmetre marcant les RPM de la velocitat 1 .....	36
<b>Figura - 7</b>	Tacòmetre marcant els RPM de la velocitat 2 .....	36
<b>Figura - 8</b>	Tacòmetre marcant els RPM de la velocitat 3 .....	37
<b>Figura - 9</b>	Matriu de Confusió per a un classificador multiclasse. Referència bibliografia: [46].....	50
<b>Figura - 10</b>	Captura de pantalla de la pestanya "Keys" de Edge Impulse per a l'obtenció de les claus API i HMAC.....	60
<b>Figura - 11</b>	Captura de pantalla missatge sortida terminal codi Python de recollida de dades per a Edge Impulse .....	61
<b>Figura - 12</b>	Captura de pantalla de la pestanya "Your devices" dintre de la plataforma Edge Impulse. ....	62
<b>Figura - 13</b>	Captura de pantalla de la pestanya "Dataset" dintre la plataforma Edge Impulse	62
<b>Figura - 14</b>	Captura de pantalla de la pestanya "Dataset" on es mostra l'alerta en el Train/Test Split dintre la plataforma Edge Impulse.....	66
<b>Figura - 15</b>	Captura de pantalla de la pestanya "Split sample" on es mostra el split realitzat Train/Test a la plataforma Edge Impulse .....	67
<b>Figura - 16</b>	Captura de pantalla de la pestanya "Dataset" on ja no es mostra l'alerta en el Train/Test Split dintre la plataforma Edge Impulse.....	68

<b>Figura - 17</b>	Captura de pantalla de la pestanya disseny general del model amb la plataforma Edge Impulse.....	70
<b>Figura - 18</b>	Captura de pantalla del llistat de learning blocks disponibles a Edge Impulse.	70
<b>Figura - 19</b>	Captura de pantalla de la configuració dels paràmetres pel filtratge i l'anàlisi espectral de les dades amb Edge Impulse. ....	74
<b>Figura - 20</b>	Gràfic de dispersió de les dades preprocessades per a cada classe amb Edge Impulse. ....	75
<b>Figura - 21</b>	Paràmetres per a l'entrenament del model d'AP amb la plataforma Edge Impulse. ....	78
<b>Figura - 22</b>	Matriu de confusió del model d'AP amb Edge Impulse.....	79
<b>Figura - 23</b>	Captura de pantalla del gràfic de dispersió de les dades ja entrenades del model d'AP amb Edge Impulse .....	81
<b>Figura - 24</b>	Pes i posició del cordill de plom afegit al ventilador per crear una anomalia.	82
<b>Figura - 25</b>	Captura de pantalla de la pestanya Dataset amb els valors de l'anomalia recol·lectats a Edge Impulse. ....	83
<b>Figura - 26</b>	Captura de pantalla dels paràmetres d'entrenament del model amb l'anomalia a Edge Impulse.....	84
<b>Figura - 27</b>	Captura de pantalla de la matriu de confusió incloent les classes de l'anomalia en Edge Impulse.....	85
<b>Figura - 28</b>	Captura de pantalla de la matriu de confusió (testing) incloent les classes de l'anomalia en Edge Impulse.....	85



## **Índex de taules.**

<b>Taula - 1</b>	Matriu de confusió binaria .....	28
<b>Taula - 2</b>	RPM de cada velocitat del ventilador .....	35
<b>Taula - 3</b>	Informe de classificació del model d'AA .....	48
<b>Taula - 4</b>	Matriu de confusió del model d'AA .....	49

## 1. Glossari de termes.

IA	Intel·ligència artificial
AA	Aprenentatge automatic
ANN	Xarxas Neuronals Artificials
AP	Aprenentatge profund
API	application programming interfaces
CNN	Xarxes neuronals convolucionals.
FFT	Fast Fourier Transformation
FIR	Finite Impulse Response
FN	False negative (Fals negatiu)
FP	False positive (fals positiu)
GND	Ground (Terra)
HMAC	Hash-based Message Authentication Code
Hz	Hertz
I2C	Inter-Integrated Circuit
IA	Intel·ligència artificial
IIR	Infinite Impulse Response
JSON	Java script object notation
macro avg	Mitjana Macro
MLP	Perceptrons multicapa
NLP	Natural Language Processing

## VIII

PCA	Anàlisi de components principals
pkl	pickle
ReLU	Unitat lineal rectificada
RNN	Xarxes neuronals recurrents.
RPM	Revolucions per minut.
SBC	Computadora de placa única
SCL	Serial Clock Line
SDA	Serial Data Line
SGD	Descens de gradient estocàstic
SVM	Maquines de vectors de suport
tanh	Tangent hiperbòlica
TFG	Treball de fi de grau
TN	True negative (veritable negatiu)
TP	True positive (veritable positiu)
VCC	Voltage Continuous Current
Weighted avg	Mitjana Ponderada
IoT	Internet of Thing (internet de les coses)

## 2. Introducció

### 2.1 Context i motivació

En l'exigent i dinàmic escenari industrial actual, apareixen reptes considerables relacionats amb l'eficiència operativa, la sostenibilitat i la qualitat del producte. En aquest escenari, la cura i manteniment de la maquinària i l'equip juguen un paper transcendental, ja que la productivitat i rendibilitat de les operacions industrials estan intrínsecament lligades al rendiment i la longevitat d'aquests equips. El manteniment predictiu, com un enfocament avançat de supervisió, ha demostrat el seu valor per a enfrontar aquests desafiaments en predir defectes i anomalies abans que es presentin, permetent a les empreses optimitzar les seves operacions i reduir costos.

El progrés en el camp de la intel·ligència artificial (IA) i la creixent disponibilitat de sensors i dispositius de seguiment, han elevat l'eficiència i precisió del manteniment predictiu a noves altures. La IA ha demostrat ser extremadament útil per a analitzar dades de sensors i identificar patrons que poden assenyalar possibles problemes. En aquest marc, aquest Treball de Fi de Grau (TFG) té com a objectiu estudiar la possibilitat de desenvolupar i implementar un model de IA que realitzi el manteniment predictiu, amb la capacitat de detectar i classificar les diferents velocitats del funcionament normal d'un ventilador TROTEC TVE 14 i un tipus d'anomalia en el mateix. Aquest objectiu s'abordarà utilitzant tècniques d'aprenentatge automàtic (AA) i aprenentatge profund (AP).

### 2.2 Objectius i abast

L'objectiu principal d'aquest TFG és estudiar la possibilitat d'implementar un sistema de manteniment predictiu basat en intel·ligència artificial (IA) en un ventilador TROTEC TVE 14, d'aquesta manera, es vol analitzar si la intel·ligència artificial pot respondre eficientment en aquesta tasca i quins reptes presenta. Per arribar a aquest propòsit es desenvoluparan dos models basats en aprenentatge automàtic (AA) i aprenentatge profund (AP). Aquests es dissenyaran per detectar, a partir de les vibracions del ventilador, les diferents velocitats normals i una possible anomalia del funcionament. Per aconseguir les dades que alimentaran els models, s'utilitzarà un acceleròmetre Adafruit ADXL345 i una Raspberry Pi 4, totes dues

es faran servir per recol·lectar les dades de les vibracions del ventilador. Els models d'IA es desenvoluparan fent servir les llibreries ScikitLearn i TensorFlow (amb la plataforma Edge Impulse).

### **2.3 Estructura del TFG**

Aquest TFG està organitzat en diferents capítols que en conjunt expliquen tots els processos duus a terme per estudiar la possibilitat de desenvolupar un sistema de manteniment predictiu d'un ventilador mitjançant intel·ligència artificial. Primerament, s'explicaran els fonaments teòrics essencials per entendre de què s'està parlant en els següents apartats. Addicionalment, s'explicaran les connexions físiques necessàries per implementar el sistema d'adquisició de dades i els càlculs matemàtics per determinar la freqüència de mostreig de les dades de l'acceleròmetre. També es comentarà el desenvolupament del codi en Python per a la recopilació de dades en temps real.

Posteriorment, s'explicarà el pas a pas del desenvolupament dels dos models d'intel·ligència artificial (AA i AP) per tal de comparar-los i veure quina és la millor solució per complir l'objectiu del TFG. Aquest procés també inclou el pas a pas de la creació del codi pel preprocessament de les dades previ a l'entrenament del model d'aprenentatge automàtic. En el cas del model d'aprenentatge profund, s'explicarà com s'ha fet servir la plataforma Edge Impulse per arribar al model.

També s'estudiaran els resultats de les prediccions dels models d'AA i AP, i s'implementarà una anomalia en el model que doni millors resultats.

Per no sobrecarregar el TFG s'han adjuntat els codis de programació en Python i les fixes tècniques dels diferents hardwares als annexos.

## 3. Fonaments teòrics

### 3.1 Manteniment

El terme manteniment es refereix a tot un conjunt de tasques que es duen a terme amb la intenció que uns certs sistemes, equips o infraestructures puguin funcionar de la millor manera possible i de manera segura. Aquest conjunt de tasques pot ser bastant ampli i variat, i incloure des de les inspeccions habituals, passant per reparacions de diferents tipus, fins als reemplaçaments necessaris, ajustos diversos i la neteja. Tot això amb l'objectiu de garantir que el funcionament sigui l'òptim i que la seguretat mai es vegi compromesa.

#### 3.1.1 Tipus de manteniment

Existeixen tres tipus principals de manteniment:

- **Manteniment correctiu:** aquest tipus de manteniment es duu a terme després que un equip hagi experimentat una avaria. És reactiu i es realitza per a reparar o reemplaçar components danyats i retornar l'equip al seu estat operatiu normal. El manteniment correctiu pot ser costós i causar temps d'inactivitat prolongats, ja que generalment es fa quan l'equip ja ha fallat.
- **Manteniment preventiu:** el manteniment preventiu és una estratègia que es caracteritza per la seva proactivitat, buscant preveure i anticipar-se a possibles avaries que puguin presentar-se en els equips. Aquest tipus de manteniment es realitza seguint un calendari o pla establert, que contempla diferents tasques que podrien incloure la lubricació de parts, l'ajust de peces, la neteja en profunditat, i el reemplaçament de components que estiguin desgastats. La idea principal de fer tot això és reduir la probabilitat d'enfrontar-se a avaries inesperades, a més d'aconseguir que els nostres equips tinguin una major vida útil.
- **Manteniment predictiu:** el manteniment predictiu es presenta com una evolució avançada de les estratègies de manteniment, fonamentant-se en la supervisió en temps real de l'estat dels equips i l'anticipació a possibles avaries abans que succeeixin. Empra una varietat d'eines i tecnologies diagnòstiques, entre les quals s'inclouen sensors, anàlisis de vibracions, examen de l'oli, termografia i algorismes

d'intel·ligència artificial, per a detectar anomalies i desgast en els equips. Gràcies al manteniment predictiu, les empreses poden organitzar les tasques de manteniment amb major eficàcia i prevenir parades no programades.

### 3.1.2 Beneficis i aplicacions del manteniment predictiu en la indústria

El manteniment predictiu ofereix nombrosos beneficis en comparació amb els enfocaments de manteniment correctiu i preventiu, entre els quals destaquen:

- **Reducció del temps d'inactivitat no planificat:** en el cas del manteniment correctiu, la reparació es fa un cop la maquinària o equip ha tingut una avaria, per tant, la producció s'haurà de parar perdent temps i diners. Amb el manteniment predictiu, al contrari, les avaries es prediuen i es localitzen abans que aquestes ocorrin. D'aquesta manera es podria estalviar molt de temps en reparació d'avaries imprevistes. El temps d'inactivitat no planificat es redueix i d'aquesta manera millora la productivitat.
- **Optimització del manteniment:** un dels grans avantatges del manteniment predictiu és que, gràcies a la reducció d'imprevistos, es pot planificar i programar els recursos i així aprofitar-los al màxim. La planificació de tasques serà més eficient i es podran reduir costos i temps d'espera. Això fa que el procés de manteniment sigui òptim i que l'eficiència operativa augmenti.
- **Extensió de la vida útil de l'equip:** gràcies a la identificació anticipada d'irregularitats i desgastos en els equips, es poden adoptar accions correctives abans que es presentin avaries greus. Això resulta en un augment de la vida útil de l'equip i una reducció en les despeses associades al seu reemplaçament.
- **Millora en la seguretat:** gràcies al manteniment predictiu es poden identificar funcionaments perillosos en la maquinària, així, es possibilita prendre les mesures necessàries per evitar el risc i prevenir accidents, garantint la seguretat dels operadors.

- **Integració amb la Indústria 4.0:** el manteniment predictiu es perfila com un element central dins de la Indústria 4.0, ja que fonamenta la seva operativitat en la recollida, l'estudi i l'intercanvi de dades en temps real entre dispositius i sistemes de producció. En adoptar el manteniment predictiu, es facilita el pas cap a la digitalització i l'automatització dels processos industrials.
- **Adopció de tecnologies emergents:** el manteniment predictiu s'ajuda de tecnologies modernes per tal de ser més precís i exacte possible. La intel·ligència artificial, per exemple, és una tecnologia innovadora que aconsegueix ser més precisa que l'ésser humà en algunes matèries. Totes aquestes tecnologies emergents milloren l'efectivitat del sistema de manteniment i milloren l'eficiència operativa.
- **Reducció de costos de manteniment:** gràcies a la capacitat de prevenir errors i millorar l'eficiència del manteniment, l'aplicació del manteniment predictiu permet reduir notablement les despeses en reparació i reemplaçament de maquinària.

### 3.1.3 Tècniques i eines utilitzades en el manteniment predictiu

En un sistema de manteniment predictiu l'objectiu és detectar les possibles avaries i funcionaments defectuosos dels equips abans que aquests arribin a aparèixer. Per això, el manteniment predictiu se centra a monitorar i analitzar la maquinària i processar aquesta informació per fer prediccions sobre el seu funcionament. Per poder dur a terme aquestes tasques es fan servir diferents eines i tècniques, algunes d'elles es poden veure a continuació:

- **Sensors:** són dispositius capaços de captar informació de l'entorn físic (llum, temperatura, moviment...) i transmetre-la. Es fan servir per analitzar i revisar el funcionament dels equips i la informació que recollin es processarà per detectar possibles alteracions en aquests.
- **Anàlisi de vibracions:** la detecció de vibracions fora de l'espectre del funcionament normal d'un equip pot ser indicativa del desgast d'aquest o de problemes en els seus components interns (rodaments, engranatges, etc). És per això que dins del



manteniment predictiu és popular la tècnica de mesurar i analitzar les vibracions dels equips.

- **Anàlisi d'oli:** analitzant l'oli d'una maquinària o equip es pot obtenir informació interessant sobre el desgast de les seves parts internes. També és important fer aquesta anàlisi en els sistemes de lubricació per avaluar si és necessari canviar-lo o realitzar algun manteniment.
- **Termografia:** en aquesta tècnica s'utilitzen càmeres infraroges per a mesurar la temperatura dels equips i així poder detectar possibles sobreescalfaments o refredaments anormals en diferents àrees. Aquestes anomalies tèrmiques indicarien problemes com el desgast dels components, la falta de lubricació o el mal funcionament de l'aïllament.
- **Algorismes d'intel·ligència artificial:** l'aprenentatge automàtic i l'aprenentatge profund són algorismes d'intel·ligència artificial que es fan servir per a analitzar grans conjunts de dades recopilades per sensors i altres eines de diagnòstic mencionades anteriorment. Mitjançant els algorismes d'intel·ligència artificial podrem detectar patrons i correlacions en les dades per a predir falles i optimitzar el manteniment.

Un cop establerts aquests fonaments teòrics del manteniment predictiu, s'ha establert una base sòlida per a comprendre com funciona aquesta tècnica i com es pot aplicar en un projecte de manteniment predictiu per a un ventilador TROTEC TVE 14.

## 3.2 Intel·ligència artificial:

### 3.2.1 Concepte d'intel·ligència artificial

La intel·ligència artificial (IA) representa una àrea de recerca dins de la informàtica, l'enfocament de la qual radica en el desenvolupament de sistemes i algorismes que poden exercir activitats que anteriorment requerien la intel·ligència humana. Aquestes activitats abasten el reconeixement de patrons, la interpretació del llenguatge natural, el pensament lògic, la resolució de problemes i l'aprenentatge. La premissa fonamental de la IA és que es

pot programar una màquina perquè emuli la intel·ligència humana i, fins i tot en determinats casos, perquè la superi en uns certs àmbits.

### 3.2.1.1 Aprenentatge supervisat, no supervisat i per reforç

Dins de l'aprenentatge de la intel·ligència artificial, es poden trobar tres classes dominants:

- **Aprenentatge supervisat:** En aquesta categoria de IA l'algorisme és entrenat utilitzant un conjunt de dades etiquetades. Les dades etiquetades són dades identificades amb una etiqueta o resultat relacionat. Així, l'objectiu de l'aprenentatge supervisat és que l'algorisme pugui predir l'etiqueta correcta a partir de les característiques apreses en l'entrenament. Exemples comuns d'algorismes d'aprenentatge supervisat inclouen la regressió lineal, l'anàlisi discriminant lineal i les màquines de vectors de suport.
- **Aprenentatge no supervisat:** en aquest tipus d'aprenentatge les dades no estan etiquetades, per tant, l'algorisme no pot fer-les servir per identificar patrons. Ha d'aprendre a relacionar les dades sense l'ajuda de l'etiqueta. Aquest tipus d'intel·ligència artificial aconsegueix agrupar les dades desconegudes (clustering). L'algorisme k-means i l'anàlisi de components principals (PCA) són exemples d'aquest tipus d'aprenentatge.
- **Aprenentatge per reforç:** L'aprenentatge per reforç és una tercera classe d'intel·ligència artificial on no es fan servir dades etiquetades i el model aprèn per si mateix, però de manera diferent de l'aprenentatge no supervisat. Els protagonistes en aquest tipus d'aprenentatge són un Agent (el model que es vol entrenar) i l'entorn on aquest interactua. D'aquesta manera, l'agent aprèn a prendre decisions interactuant amb l'entorn. Mitjançant un sistema de premis i penalitzacions, l'agent es retroalimenta i aprèn a aconseguir les màximes recompenses possibles. Un símil d'aquest tipus d'aprenentatge a la vida real serien els premis que se li donen a un gos per educar-lo. Aquest tipus d'aprenentatge s'utilitza sovint en aplicacions com el control de robots i la presa de decisions en jocs on el rival és una màquina.

### **3.2.1.2 Aprenentatge automàtic (machine learning) vs. aprenentatge profund (deep learning)**

Dins de la intel·ligència artificial es troben dos enfocaments diferents: l'aprenentatge automàtic i l'aprenentatge profund. Tots dos estan basats a desenvolupar algorismes capaços d'aprendre a partir de dades.

- **Aprenentatge automàtic:** Aquesta forma de IA, mitjançant algorismes matemàtics i estadístics, ensenya a les màquines a aprendre de les dades i millorar el seu rendiment en una missió específica. En els algorismes dins d'aquest aprenentatge es poden trobar mètodes de regressió, classificació i agrupació, que poden ser supervisats o no. Aquest sistema es fa servir en aplicacions d'anàlisi de dades, predicció, detecció d'anomalies i presa de decisions.
- **Aprenentatge profund:** és una subcategoria dins de l'aprenentatge automàtic. La ciència de l'aprenentatge profund està basada en la biologia del cervell de l'home i fa servir una sèrie de capes de neurones artificials per analitzar les dades d'una manera semblant a la humana. Això ho fa amb xarxes neuronals artificials (ANN pel seu nom en anglès), uns algorismes per nivells que processen les dades. L'aprenentatge profund ha demostrat ser una alternativa molt potent, capaç de processar imatges, textos i dades en diferents suports audiovisuals.

L'aprenentatge profund processa les dades d'una manera més abstracta que l'aprenentatge automàtic. Mentre que a l'AA es requereix la intervenció d'una persona humana que processa les dades, l'aprenentatge profund pot aprendre per si mateix gràcies a la retroalimentació i l'actualització dels pesos (un paràmetre que mesura la importància de cada característica) durant l'entrenament del model.

Aquestes formes d'IA són complementàries i fan servir algorismes per processar les dades. L'aprenentatge automàtic utilitza càlculs matemàtics i el profund les mencionades ANN. Tots dos tenen els seus avantatges i desavantatges i la decisió de fer servir un o altra dependrà del problema a resoldre.

### 3.3 Algorismes, xarxes i funcions per a l'aprenentatge automàtic i l'aprenentatge profund:

Els algorismes, en conjunt amb les xarxes i funcions, són la base de tot el funcionament dels models d'aprenentatge d'intel·ligència artificial. És important tenir algunes nocions sobre aquests elements tan presents en l'aprenentatge profund i automàtic.

#### 3.3.1 Algorismes d'aprenentatge automàtic:

Els algorismes en els quals estan basats els models d'aprenentatge automàtic poden pertànyer a diferents classes: regressió, classificació, arbres de decisió, màquines de vectors de suport, etc.

- **Regressió:** una tècnica que calcula la relació entre una variable dependent i una o més variables explicatives. Dins de la regressió es poden trobar la regressió lineal i la regressió logística.
- **Classificació:** es basa a etiquetar i categoritzar les dades/observacions en funció de les seves característiques. Els algorismes de classificació més coneguts són k-NN (k-veïns més pròxims), arbres de decisió i màquines de vectors de suport.
- **Arbres de decisió:** també és un concepte conegut fora del món de la intel·ligència artificial. Són mapes que es fan servir per prendre decisions. Comencen plantejant una pregunta, que es va ramificant amb possibles resultats, per això es diuen arbres. En el món de la IA aquests són algorismes que tenen un funcionament semblant al dels mapes, es fan servir per analitzar dades i predir resultats a partir d'algunes condicions.
- **Màquines de vectors de suport (SVM):** són algorismes d'aprenentatge supervisat amb l'objectiu de crear un hiperplà amb el marge més ample possible que separi el millor que pugui punts de dades de dues classes.

### 3.3.2 Xarxes neuronals i arquitectures d'aprenentatge profund

L'aprenentatge profund utilitza xarxes neuronals artificials, és a dir, estructures d'algorismes per nivells que pretenen analitzar la informació amb una lògica semblant a la humana. D'aquesta manera, aquestes xarxes amb múltiples capes aprenen representacions jeràrquiques de les dades. Existeixen diverses arquitectures de xarxes neuronals, com perceptrons multicapa (MLP), xarxes neuronals convolucionals (CNN) i xarxes neuronals recurrents (RNN).

- **Perceptrons multicapa (MLP):** són xarxes neuronals feedforward, és a dir unidireccionals, que consisteixen en una capa d'entrada, una o més capes en el mig (anomenades capes ocultes) i una capa de sortida. Els MLP s'utilitzen en una àmplia gamma d'aplicacions, com la classificació, regressió i aproximació de funcions.
- **Xarxes neuronals convolucionals (CNN):** aquestes arquitectures d'aprenentatge destaquen per la seva similitud al funcionament de les neurones en el còrtex visual a causa del processament de dades en forma de reixa. Destaquen especialment en el processament d'imatges. Mitjançant aquestes capes, apliquen filtres locals a les dades d'entrada i aprenen característiques espacials i de forma, d'aquesta manera, un filtre processa les vores, un altre els colors, etc.
- **Xarxes neuronals recurrents (RNN):** són arquitectures d'aprenentatge profund dissenyades per a processar seqüències de dades. Les RNN són molt útils per dades de sèries temporals, ja que tenen una mena de memòria interna. Es fan servir sovint en Natural Language Processing (NLP).

### 3.3.3 Funcions d'activació, optimització, funció de pèrdua i regularització:

Tant l'aprenentatge automàtic com el profund fan servir funcions matemàtiques en diferents punts del procés. Comprendre aquestes funcions és essencial per poder crear models que puguin processar la informació i complir el seu objectiu. Cada funció té un rol específic en l'entrenament dels models d'aprenentatge, sempre amb la finalitat d'optimitzar les prediccions i aconseguir resultats precisos.

- **Funcions d'activació:** aquestes són funcions no lineals aplicades a les sortides de les neurones en una xarxa neuronal. Aquestes funcions permeten que les xarxes neuronals aprenguin relacions més complexes entre les dades. Algunes funcions d'activació comunes inclouen la funció sigmoide, la tangent hiperbòlica (tanh) i la unitat lineal rectificada (ReLU).
- **Optimització:** l'objectiu en l'optimització és que la xarxa neuronal sigui el més funcional i precisa possible, en aquest procés s'ajusten els pesos perquè els resultats que doni el model es desviïn el mínim de la realitat. Entre els algorismes més coneguts d'aquest tipus es poden trobar el descens de gradient estocàstic (SGD), el moment, Adagrad, RMSprop i Adam.
- **Funció de pèrdua:** és una funció que compara les prediccions del model d'aprenentatge profund amb els resultats reals i mostra la distància entre aquests. És a dir, demostra quant de desviada és la resposta donada pel model de la realitat. L'objectiu de l'Optimització, que ha sigut mencionada abans, és minimitzar aquesta pèrdua. Exemples de funcions de pèrdua inclouen l'entropia creuada per a problemes de classificació i l'error quadràtic mitjà per a problemes de regressió.
- **Regularització:** tècnica que es fa servir per evitar el sobreajust en els models d'aprenentatge (quan el model està "sobrentrenat" per les dades que ja coneix però és incapaç de fer prediccions correctes amb noves dades). Dins de la regularització hi ha tècniques diferents, L1 i L2. La L1 per exemple funciona amb un sistema de penalització. Una altra tècnica seria el dropout (abandó) i el que fa és desconnectar algunes neurones en cada capa de la xarxa neuronal.

### 3.4 Processament de senyals i extracció de característiques

El processament de senyals i l'extracció de característiques són dos aspectes d'alta importació en el món de la intel·ligència artificial. El processament de senyals, s'encarrega únicament de transformar senyals d'entrada en un format que sigui interpretable pels models de IA. Per altra

banda, l'extracció de característiques se centra a identificar les característiques més rellevants dels senyals per tal de facilitar la digestió al model IA.

### 3.4.1 Conceptes bàsics de processament de senyals:

- **Mostreig:** en aquest pas es prenen les dades de l'objecte que s'estudiarà i es converteixen a un format que es pugui llegir digitalment, per exemple en el cas d'aquest TFG, el mostreig es faria en el ventilador. L'acceleròmetre mesura les acceleracions dels tres eixos del ventilador (X,Y i Z) i el senyal analògic del ventilador es convertirà en un senyal digital.
- **Transformada de Fourier:** Mitjançant aquesta tècnica matemàtica es transforma un senyal del domini del temps al domini de la freqüència. En aquest projecte interessa calcular la Transformada Discreta de Fourier, ja que aquesta és la que es fa servir per detectar senyals discrets en el temps, com són les dades digitals. L'algorisme eficient per calcular la Transformada Discreta de Fourier serà La Transformada Ràpida de Fourier (FFT) En aquest projecte, s'aplicarà la FFT als senyals d'acceleració per a identificar components freqüencials i caracteritzar el comportament del ventilador.
- **Filtratge:** com el seu nom indica, es tracta de filtrar les dades perquè siguin tan pures com sigui possible. Les dades que podrien venir d'agents externs (com per exemple soroll o vibracions d'objectes diferents de l'estudiat durant la recollida de dades) s'eliminen o suavitzen. En aquest TFG es farà ús d'aquesta tècnica per afinar el senyal que es recull del ventilador i eliminar freqüències més altes que podrien venir d'agents externs.

### 3.4.2 Extracció de característiques:

Aquest procés, tal com es pot deduir pel seu nom, se centra a treure les qualitats de les dades i convertir la informació recollida en dades útils pel model d'aprenentatge. Així, es passa de tenir unes dades brutes a tenir una representació resumida de les dades recollides. En el manteniment predictiu aquest procés és molt important perquè ajuda als algorismes a llegir la informació correctament. Les característiques identificades seran les entrades que agafaran els models d'aprenentatge pel seu entrenament.

- **Estadístiques bàsiques:** en el procés d'extracció de característiques, un dels primers passos és el càlcul d'estadístiques bàsiques que puguin resumir la informació recollida. Proporcionaran informació general sobre els senyals que s'estan mostrejant i, gràcies a això, serà també més fàcil veure anomalies o comportaments estranys en el funcionament de l'equip. Algunes d'aquestes serien la mitjana, la variància, la desviació estàndard, el rang interquartílic i el coeficient de variació.
- **Característiques en el domini de la freqüència:** les característiques de les dades també es poden extreure en el domini de la freqüència, és a dir, en una representació freqüencial de les dades. Per això s'ha d'aplicar la FFT que ja s'ha explicat en l'apartat [3.4.1]. Un cop més, aquesta informació pot ser útil en el manteniment predictiu per detectar anomalies. Així, s'extrauran característiques com la freqüència fonamental, les amplituds dels components freqüencials, l'energia espectral i la densitat espectral de potència.
- **Característiques temporals:** a part de les característiques en el domini de la freqüència que s'han mencionat en el punt anterior, també són útils les característiques extretes en el domini del temps. Aquestes aporten informació sobre la dinàmica en el temps del senyal analitzat i les estructures de repetició d'aquest. Exemples d'aquestes característiques són l'autocorrelació, l'entropia del senyal, la taxa d'encreuaments per zero i la taxa d'encreuaments per nivell.

## 3.5 Raspberry Pi i acceleròmetres

### 3.5.1 Raspberry Pi 4: característiques, especificacions i funcionalitats

La Raspberry Pi 4 és una computadora de placa única (SBC) desenvolupada per la Fundació Raspberry Pi. A més de ser compacta i econòmica, la placa de la Raspberry Pi 4 és un bon recurs per a aplicacions en sistemes encastats, educació, robòtica, etc. Ofereix una sèrie de millores significatives en comparació amb les seves predecessores, entre altres, majors capacitats de processament, memòria i connectivitat.



Algunes de les principals característiques i especificacions, que es poden trobar en l'annex [X], de la Raspberry Pi 4 inclouen:

- Processador Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) de 64 bits a 1.5 GHz
- Memòria LPDDR4-3200 SDRAM de 2 GB, 4 GB o 8 GB (depenent del model)
- Connectivitat sense fil Wi-Fi 802.11ac i Bluetooth 5.0
- Ethernet Gigabit
- 2 ports USB 3.0 i 2 ports USB 2.0
- Sortida de vídeo HDMI de doble pantalla i resolució 4K
- Port CSI per a cambra i port DSI per a pantalla tàctil
- Ranura per a targeta microSD per a emmagatzematge i càrrega del sistema operatiu

Per a aconseguir el propòsit d'aquest TFG, la Raspberry Pi 4 serà essencial. Aquesta farà de plataforma de control i computació, on es rebran les dades de l'acceleròmetre Adafruit ADXL345 i on s'executaran els models d'aprenentatge encarregats de fer les prediccions de les classes correctament.

### **3.5.2 Acceleròmetres: funcionament, tipus i aplicacions, amb èmfasis en el Adafruit ADXL345**

És essencial esmentar els acceleròmetres, uns sensors particulars que s'encarreguen de mesurar l'acceleració. Aquesta acceleració pot ser estàtica, produïda per la gravetat, o dinàmica, originada pel moviment o les vibracions. En aprofundir en els acceleròmetres, ens trobem amb diversos tipus, cadascun amb el seu propi principi de funcionament i amb una gamma d'aplicacions. Entre ells, destaquen els acceleròmetres piezoelèctrics, capacitius i d'efecte Hall.

En el marc d'aquest TFG, es recorre a l'acceleròmetre Adafruit ADXL345, una peça tecnològica notable per la seva funció de sensor d'acceleració digital en tres dimensions. El seu funcionament se sustenta en la tecnologia de microelectromecànica de capacitància, coneguda per les seves sigles en anglès com MEMS. La utilitat de l'ADXL345 radica en la seva capacitat per a proporcionar mesures d'acceleració en els eixos X, Y i Z amb un nivell de resolució elevat i un soroll mínim. Algunes de les característiques, que es poden trobar a l'annex [XI], de l'ADXL345 inclouen:

- Rang de mesurament:  $\pm 2$ ,  $\pm 4$ ,  $\pm 8$  o  $\pm 16$  g
- Resolució de 13 bits (fins a 3.9 mg/LSB)
- Interfície digital SPI o I2C
- Filtres digitals programables
- Consum d'energia ultra baix

En aquest TFG, es fa servir l'acceleròmetre ADXL345 com una eina essencial per a analitzar les vibracions del ventilador TROTEC TVE 14. Aquest enfocament permetrà detectar i classificar el funcionament normal i una anomalia de l'equip. Això serà possible gràcies a la detallada anàlisi i processament dels senyals de vibració que proporciona aquest l'acceleròmetre.

### 3.5.3 Protocol de comunicació I2C

La comunicació I2C és un mètode per connectar dispositius digitals i que s'envii informació entre ells. Els dispositius que coordinen la comunicació són els mestres i es comuniquen amb els dispositius esclaus. Aquest tipus de comunicació es fa servir molt amb sensors (com seria el cas de l'acceleròmetre Adafruit d'aquest TFG), ja que és simple i molt útil. La instal·lació requereix poc cablejat pel que facilita les connexions i resulta molt útil en espais petits on la simplicitat del cablejat s'agraeix especialment.

Algunes característiques clau del protocol I2C inclouen:

- **Comunicació mestre-esclau:** un dispositiu mestre (per exemple, Raspberry Pi 4) pot comunicar-se amb un o diversos dispositius esclaus (per exemple, acceleròmetres) en la mateixa línia de dades.
- **Direccions de 7 o 10 bits:** els dispositius esclaus de les connexions I2C tenen una direcció de 7 o 10 bits que és única i no pot haver-hi cap altre dispositiu esclau amb la mateixa direcció. Gràcies a això el mestre es pot comunicar amb els esclaus sense problemes.

- **Transmissió bidireccional de dades:** tots els dispositius estan connectats als mateixos cables, un cable SDA i un cable SCL. Així es simplifiquen les connexions i es poden transmetre les dades en totes dues direccions.
- **Velocitats de comunicació variables:** I2C permet una velocitat de comunicació de 400 kilobits per segon (kbits/s) en modalitat ràpida (fast-mode), 1 mega bit per segon (Mbit/s) en modalitat ràpid plus i fins a 3.4 Mbits/s en alta velocitat (High-speed mode).

En aquest projecte, el protocol I2C s'utilitza per a connectar la Raspberry Pi 4 a l'acceleròmetre Adafruit ADXL345 i transmetre dades de mesurament d'acceleració entre els dispositius. La Raspberry Pi 4 actua com a mestre, mentre que l'ADXL345 és l'esclau. La implementació de l'I2C en la Raspberry Pi 4 permet una fàcil integració i configuració de l'acceleròmetre en el sistema de manteniment predictiu i garanteix una comunicació ràpida i de confiança per a l'adquisició de dades en temps real.

### 3.6 Llibreries i eines de programació

Per desenvolupar el sistema de manteniment predictiu basat en IA que es vol estudiar en aquest TFG seran necessàries una sèrie de llibreries i eines per la programació. En aquest apartat s'explicaran unes nocions de Python, TensorFlow, Scikit-learn i Edge Impulse per entendre alguns dels processos que s'expliquen més endavant.

#### 3.6.1 Python

Python és un dels llenguatges de programació més utilitzats en el món, és de propòsit general, és a dir, està dissenyat per resoldre problemes de tota classe. Python és un llenguatge molt versàtil i gràcies a la seva simplicitat es fa servir àmpliament en el món de la intel·ligència artificial. A continuació es poden veure alguns dels avantatges d'aquest llenguatge:

- **Facilitat d'aprenentatge:** Python és un llenguatge amb una sintaxi relativament senzilla, similar a la de l'anglès, això fa que sigui un codi raonablement fàcil d'aprendre.

- **Gran quantitat de llibreries:** aquest llenguatge compta amb una gran varietat de llibreries de machine learning i deep learning, com TensorFlow, Keras, Scikit-learn, entre altres. Gràcies a això el desenvolupament i la implementació de solucions d'intel·ligència artificial es pot aplicar de manera més simple.
- **Comunitat activa:** com que Python és un llenguatge tan popular, hi ha una gran quantitat d'usuaris i desenvolupadors en constant creació i actualització de llibreries i eines relatives a l'ús del codi Python que estan a l'accés de tothom.

### 3.6.2 TensorFlow

TensorFlow és una biblioteca de programari de codi obert creada per Google Brain Team. Aquesta llibreria permet l'aplicació d'algorismes d'aprenentatge profund en aplicacions d'intel·ligència artificial i proporciona una plataforma flexible i eficient per a la manipulació de dades.

En aquest treball, TensorFlow s'utilitza per a implementar models d'aprenentatge profund per a l'anàlisi i la classificació de dades d'acceleració del ventilador TROTEC TVE 14 conjuntament amb la plataforma Edge Impulse.

### 3.6.3 Scikit-learn

Scikit-learn és una llibreria de Python enfocada a la intel·ligència artificial i de codi obert, és a dir, gratuïta. Compta amb una àmplia varietat d'algorismes per l'aprenentatge supervisat i no supervisat i pel preprocessament de dades entre altres. En aquest TFG la llibreria Scikit-learn es fa servir en el primer model de predicció que es desenvolupa, un model d'aprenentatge automàtic que pretén detectar les diferents velocitats del ventilador en funcionament.

### 3.6.4 Edge Impulse

Edge Impulse és una plataforma online d'intel·ligència artificial amb la qual es poden desenvolupar models d'aprenentatge automàtic i profund. Amb una interfície "user friendly"

Edge Impulse permet a l'usuari crear models que recol·lectin les dades pel seu posterior entrenament en el núvol. Edge Impulse està dissenyat per treballar amb tecnologia Edge, és a dir, dispositius que poden analitzar i processar les dades en temps reals sense la necessitat d'un servidor d'internet. La Raspberry Pi, per exemple, és un dispositiu edge.

En aquest TFG Edge Impulse es farà servir per crear un model d'aprenentatge profund capaç de detectar les diferents velocitats del ventilador TROTEC TVE 14, per aconseguir això, també es recol·lectaran les dades amb l'acceleròmetre Adafruit connectat a la Raspberry Pi 4. Després, es farà la validació i testing per avaluar la qualitat de les prediccions del model.

En resum, Python, TensorFlow, Scikit-learn i Edge Impulse són eines i llibreries fonamentals en el desenvolupament d'aquest projecte de manteniment predictiu per al ventilador TROTEC TVE 14 utilitzant intel·ligència artificial. Aquestes llibreries i eines proporcionen una àmplia opció d'algorismes, funcions i arquitectures predefinides per a facilitar la implementació i avaluació de models d'aprenentatge automàtic i aprenentatge profund, així com el desplegament de solucions d'intel·ligència artificial en dispositius edge com la Raspberry Pi 4.

## 3.7 Validació i avaluació de models d'intel·ligència artificial

### 3.7.1 Mètriques de rendiment

Després de l'entrenament d'un model d'aprenentatge automàtic o profund s'ha de mesurar l'èxit d'aquest, per això es fa un procés d'avaluació o validació, on es treuen algunes estadístiques per observar com de precises són les prediccions. Aquests indicadors es diuen mètriques de rendiment. En aquest TFG les mètriques de rendiment que s'han fet servir per comprovar el rendiment dels models de IA són les següents:

- **Exactitud (Accuracy):** l'exactitud és la proporció de resultats correctes entre tots els resultats que prediu el model. En aquest cas, s'ha de diferenciar si s'està fent una classificació binària o multiclasse.

En el cas d'una classificació binària, el model ha de predir una única condició (per exemple, el ventilador està en funcionament o no, és a dir, està en velocitat 0 o no). En

aquest cas l'exactitud es calcula dividint la suma dels resultats correctes entre tots els resultats. Els resultats correctes serien els veritables positius (True positive 'TP') i els veritables negatius (True Negative 'TN'). Per calcular el total de resultats als TP i TN se'ls hauria de sumar els falsos positius (False Positive 'FP') i falsos negatius (False Negative 'FN').

D'aquesta manera, la formula per calcular l'exactitud en models de classificació binària és:

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN}$$

En el cas d'una classificació multiclasse, és a dir, on el model ha de predir diverses condicions (per exemple, el ventilador està funcionant en Velocitat 0, 1, 2 o 3) es fa una matriu de confusió (que veurem en els següents punts) més amplia i el càlcul de l'exactitud es pot complicar. Per això l'exactitud en casos de classificació multiclasse es calcula dividint els resultats o prediccions correctes entre tots els resultats o prediccions.

$$Exactitud = \frac{\text{Prediccions Correctes}}{\text{Total Prediccions}}$$

O expressada d'una altra manera però amb el mateix significat:

$$Exactitud = \frac{\sum TP}{\sum TP + \sum FP + \sum FN}$$

- **Precisió:** la precisió se centra a calcular la proporció dels veritables positius (TP) entre totes les prediccions positives que s'han fet. Per exemple, en el cas d'aquest TFG la precisió de la velocitat 0 del ventilador seria el càlcul de les prediccions correctes que s'han fet indicant que el ventilador estava en velocitat 0, entre totes les prediccions (correctes i incorrectes) en les que el model ha dit que el ventilador estava en aquesta velocitat. D'aquesta manera, la precisió mesura la capacitat del model de detectar correctament les velocitats o anomalies del ventilador.

$$Precisió = \frac{TP}{TP + FP}$$

- **Recall (Sensibilitat):** el Recall o Sensibilitat (es fa servir més el terme Recall a l'ambient científic) se centra a identificar quin percentatge de casos positius reals s'han identificat. Per exemple, i continuant amb el cas de la velocitat 0, el Recall seria la proporció de casos de velocitat 0 correctament detectats entre tots els casos en els quals el ventilador estave realment en aquesta velocitat. És a dir, aquesta mètrica mesura la capacitat per detectar correctament les classes.

$$Recall = \frac{TP}{TP + FN}$$

- **F1-score:** és una mesura harmònica de la precisió i la sensibilitat (Recall), que combina totes dues mètriques en un únic valor. La F1-score és especialment útil quan la distribució de classes (etiquetes) és desequilibrada, ja que en tenir en compte tots dos valors, només sortirà amb un alt valor d'F1-score si tots dos, sensibilitat i precisió, són alts.

$$F1 - score = \frac{2 \cdot (Precisió \cdot Recall)}{Precisió + Recall}$$

- **Matriu de confusió:** en la Matriu de confusió s'expressen en una única taula tots els resultats, les prediccions que ha fet el model, i els valors reals. Amb aquesta mètrica es pot calcular el rendiment del model, ja que indica totes les prediccions correctes i incorrectes que ha fet.

	Predicció Positiva	Predicció Negativa
Clase Positiva	Veritables Positius	Falsos Negatius
Clase Negativa	Falsos Positius	Veritables Negatius

**Taula - 1** Matriu de confusió binaria

Cada cel·la de la matriu representa un tipus de predicció (veritable positiu, fals positiu, veritable negatiu o fals negatiu) i permet avaluar el rendiment del model a través de les mètriques esmentades anteriorment.

- **Suport:** es refereix al nombre total d'instàncies reals de cada classe, és a dir, cada registre o dada recollida en cada classe.
- **Mitjana Macro i Ponderada:** mentre que la precisió, el Recall i la F1-Score es calculen classe per classe, les mitjanes Macro i Ponderada fan una mitja d'aquestes mètriques tenint en compte totes les classes. La mitjana Macro fa el càlcul sense tenir en compte la desigualtat entre classes, és a dir, totes les classes es consideren iguals. Al contrari, la Mitjana Ponderada calcula les mètriques en cada classe i fa la mitjana tenint en compte el nombre d'instàncies, és a dir, el suport. D'aquesta manera, la Mitjana Ponderada té en compte la desigualtat entre etiquetes.

$$\text{Mitjana Macro Precisió} = \frac{\sum \text{Precisió (Classe } n)}{\text{total de classes}}$$

$$\text{Mitjana Macro Recall} = \frac{\sum \text{Recall (Classe } n)}{\text{total de classes}}$$

$$\text{Mitjana Macro F1 - Score} = \frac{\sum \text{F1 - Score (Classe } n)}{\text{total de classes}}$$

$$\text{Mitjana Ponderada Precisió} = \frac{\sum (\text{Precisió (Clase } n) \cdot \text{suport (Clase } n))}{\text{suport total}}$$

$$\text{Mitjana Ponderada Recall} = \frac{\sum (\text{Recall (Clase } n) \cdot \text{suport (Clase } n))}{\text{suport total}}$$

$$\text{Mitjana Ponderada F1 - Score} = \frac{\sum (\text{F1 - Score (Clase } n) \cdot \text{suport (Clase } n))}{\text{suport total}}$$



## **4. Implementació del sistema d'adquisició de dades**

### **4.1 Descripció general del sistema d'adquisició de dades**

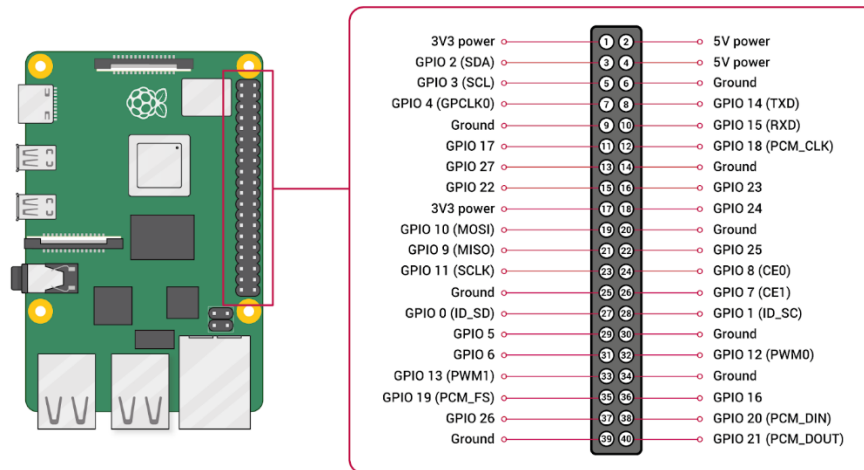
El sistema mitjançant el qual s'adquireixen les dades que recull el dispositiu (en aquest cas l'acceleròmetre Adafruit ADXL345) és fonamental, ja que aquest sistema és responsable de recopilar els senyals de l'acceleròmetre i transmetre'ls a la Raspberry Pi 4 per al seu posterior processament i anàlisi. A continuació, es proporciona una descripció general del sistema d'adquisició de dades i la seva interacció.

### **4.2 Connexions físiques entre la Raspberry Pi 4 i l'acceleròmetre Adafruit ADXL345**

A continuació es mostren les connexions físiques entre la Raspberry Pi 4 i l'acceleròmetre Adafruit ADXL345:

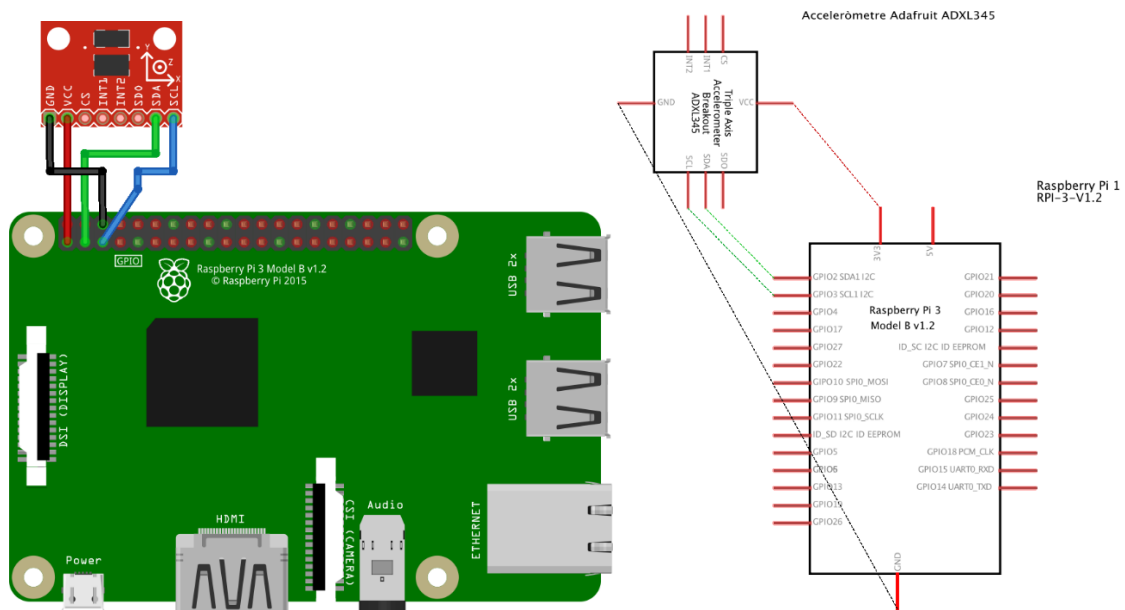
- Connectar la font d'alimentació (VCC) de l'acceleròmetre ADXL345 al pin 1 (3.3V) de la Raspberry Pi 4. L'ADXL345 pot funcionar amb voltatges de 2.0V a 3.6V, per la qual cosa és compatible amb la sortida de 3.3V de la Raspberry Pi.
- Connectar la terra (GND) de l'acceleròmetre ADXL345 al pin 6 (GND) de la Raspberry Pi 4.
- Connectar la línia de dades I2C (SDA) de l'acceleròmetre ADXL345 al pin 3 (GPIO 2, SDA1) de la Raspberry Pi 4.
- Connectar la línia de rellotge I2C (SCL) de l'acceleròmetre ADXL345 al pin 5 (GPIO 3, SCL1) de la Raspberry Pi 4.

Seguidament, es mostra de manera gràfica els diferents pins de la Raspberry Pi 4 amb les seves ubicacions i el seus respectius noms.



**Figura - 1** Ubicació pins de connexió de la Raspberry Pi 4. Imatge obtinguda de la referència [6].

Tot seguit, es mostra de manera gràfica l'esquema elèctric de les connexions dels pins entre la Raspberry Pi 4 i l'acceleròmetre Adafruit ADXL345 realitzat amb el programa Fritzing.



**Figura - 2** Esquema elèctric de les connexions entre l'ADXL345 i la Raspberry Pi 4 (imatges realitzades amb el programa Fritzing)

### **4.3 Posicionament del sensor per obtenir les vibracions òptimes**

No totes les ubicacions del ventilador són aptes per tal d'enregistrar dades rellevants per a l'entrenament dels models d'intel·ligència artificial. De fet, aquest pas, és una de les parts més importants per tal d'obtenir bons rendiments en els models. Per aquest motiu, s'han realitzat diverses proves empíriques per tal d'extreure conclusions coherents d'on posicionar l'acceleròmetre Adafruit ADXL345.

La primera prova és a escala del tacte humà, i va tractar d'anar passant el dit per diferents ubicacions del ventilador. Aquesta prova es va dur a terme amb el ventilador a velocitat tres (la màxima velocitat del ventilador en funcionament normal). Es va poder determinar que la carcassa no generava suficients vibracions quant a tacte, així que es va prendre la decisió de treure la tapa. Un cop treia la tapa, es van poder observar dues posicions on es percebien més vibracions, tal com es mostra a la figura [4]. A partir d'aquest punt, es va analitzar quina de les dues posicions podia extreure millors acceleracions per part de l'acceleròmetre. Com aquesta tasca no es pot determinar simplement pel tacte humà, es va decidir fer una segona prova per poder determinar la posició òptima.

La segona prova va consistir a connectar dos acceleròmetres al ventilador, un a cada ubicació on la prova 1 havia detectat un major grau de vibracions. La connexió dels acceleròmetres es va realitzar tal com s'explica en l'apartat [4.2]. Després d'extreure les dades en un CSV (Explicat en l'apartat [5.1]), se'n van gràfica les dades seguint l'Annex [III] per tal de poder determinar quin sensor captava millors dades. Es va poder determinar que la posició número 1 obtenia una gràfica lleugerament més clara de les dades, però no era suficient per obtenir una conclusió en ferm.

La tercera prova que es va realitzar va ser executar el model d'AA amb la llibreria de RandomForestClassifier de Scikit-learn per tal d'avaluar el model posteriorment i veure quina de les dues posicions assolia millor rendiment. En cada una de les dues posicions es van recollir les dades, fer el preprocessament i entrenar i avaluar el model d'AA. En l'avaluació del model, es va poder confirmar que la diferència de rendiment entre les dues posicions era del 2%, sent la posició 1 la millor, tal com es mostra en la figura [3].

Malgrat el petit avantatge de la posició 1 en els resultats de la tercera prova, aquesta posició implica també alguns desavantatges. Per començar, la posició 1 està situada just sobre el

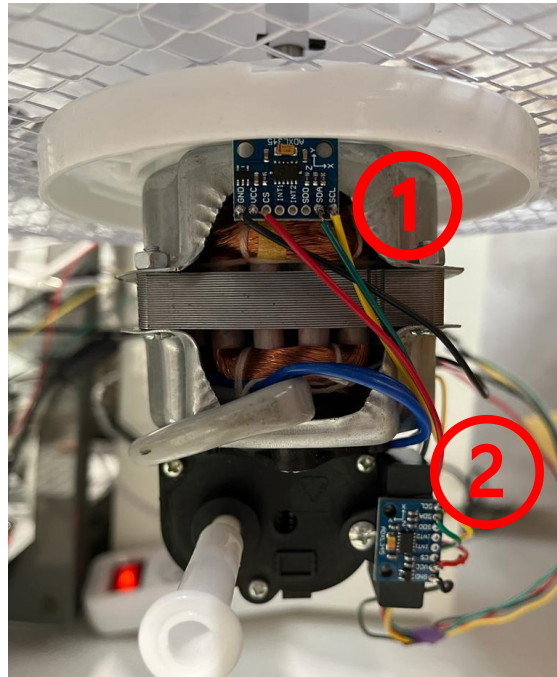
motor del ventilador, la qual cosa implica un major escalfament en el seu funcionament continuat i això podria amb el temps afectar a l'electrònica del sensor així com a l'adhesiu que enganxa aquest en el ventilador. Per una altra banda, en el cas d'avaría o manteniment del motor del ventilador, es correria el risc de malmetre el sensor durant les tasques de reparació o manteniment, sigui per accident o per la necessitat de treure'l per poder accedir a certes parts del motor.

La posició 2, al contrari, està situada en un termoplàstic extern al motor subjectat per un caragol, per la qual cosa, en el cas de necessitar acomplir tasques de manteniment, es podria desmuntar amb rapidesa i sense major problema.

D'agut a la poca diferència de rendiment en el model d'AA entre les dues posicions, i tenint en compte les raons mencionades anteriorment, he decidit finalment escollir la posició 2. Al fi i a la cap un 2% de diferència en el rendiment del model d'AA no compensa els possibles problemes físics que pugui implicar la posició 1.

Posició 1				Posició 2			
	precision	recall	f1-score		precision	recall	f1-score
0.0	0.99	1.00	0.99	0.0	0.92	0.97	0.95
1.0	0.47	0.69	0.56	1.0	0.59	0.58	0.58
2.0	0.43	0.28	0.34	2.0	0.43	0.44	0.43
3.0	0.55	0.45	0.49	3.0	0.52	0.50	0.51
accuracy			0.64	accuracy			0.62
macro avg	0.61	0.60	0.60	macro avg	0.62	0.62	0.62
weighted avg	0.64	0.64	0.63	weighted avg	0.62	0.62	0.62

**Figura - 3** Comparació exactitud, posició 1 vs. posició 2



**Figura - 4** Possibles posicionaments del sensor en el ventilador

#### **4.4 Càlcul de la freqüència de mostreig**

##### **4.4.1 Mesurament de les revolucions per minut del ventilador amb un tacòmetre làser**

Per tal d'establir una freqüència de mostreig de les dades d'acord amb el teorema de Nyquist-Shannon, que s'explica més àmpliament en l'apartat [4.4.2], era necessari mesurar les revolucions per minut (RPM) del ventilador TROTEC TVE-14, ja que en la seva fitxa tècnica, que es pot trobar en l'annex [IX], no s'especifica la freqüència màxima del motor.

Per mesurar les RPM del ventilador, va ser necessari adquirir un tacòmetre làser, pel fet que aquest dispositiu permet mesurar la velocitat de rotació d'un objecte sense necessitat de contacte físic. En aquest cas resulta especialment útil, ja que el contacte amb el ventilador pot ser perillós, impracticable, o fins i tot influir en la velocitat de rotació d'aquest.

El tacòmetre làser mesura les revolucions per minut mitjançant un feix de llum que rebota amb una petita banda reflectora adherida a l'objecte que gira. Quan l'objecte, en aquest cas el ventilador, gira, la banda reflectora es mou i la llum làser del tacòmetre es reflecteix de tornada cada cop que la banda passa per davant, calculant així la velocitat de rotació.

En aquest projecte, a causa de la naturalesa de l'objecte estudiat (un ventilador), l'ús d'un tacòmetre làser era l'opció més segura. El procés de mesurament és senzill: es col·loca una cinta reflectora a una distància determinada de l'eix de rotació (en aquest cas, a 2,5 cm, una ubicació còmoda donada la morfologia del ventilador).

No obstant això, durant els mesuraments, es van trobar algunes dificultats. El plàstic blanc del ventilador va resultar ser altament reflector, causant interferències en el mesurament, ja que el tacòmetre també detectava aquesta superfície, a més de la cinta reflectora. Per a solucionar aquest problema, es va pintar la zona afectada de negre, amb la finalitat de maximitzar l'absorció de llum. Tot i així, es va detectar un punt brillant que continuava causant interferències. Com a solució definitiva, es va col·locar un cartó negre mat en el camí del làser, la qual cosa va permetre al tacòmetre detectar només la cinta reflectora durant cada gir del ventilador.



**Figura - 5** Cartó negre amb cinta reflectant del tacòmetre

Les RPM del ventilador es van mesurar en les seves tres velocitats, obtenint els següents resultats:

Mode de velocitat	Revolucions per minut (RPM)
1	1042,5
2	1137.6
3	1251,2

**Taula - 2** RPM de cada velocitat del ventilador

A continuació, es mostra les proves gràfiques de les mesures per a cada velocitat (figures [6],[7] i [8]).



**Figura - 6** Tacòmetre marcant les RPM de la velocitat 1



**Figura - 7** Tacòmetre marcant els RPM de la velocitat 2





**Figura - 8** Tacòmetre marcant els RPM de la velocitat 3

Gràcies a aquestes mesures es va poder treure amb precisió la velocitat màxima de rotació del ventilador d'aquest projecte, i a partir d'aquesta informació es va calcular la freqüència de mostreig òptima per a captar les vibracions més característiques del seu funcionament, tal com es discutirà en la secció [4.4.2].

#### **4.4.2 Teorema de mostreig de Nyquist-Shannon**

El Teorema de Mostreig de Nyquist-Shannon és una de les peces fonamentals dins del tractament de senyals digitals. El teorema diu que per reconstruir un senyal continu, el mostreig a de ser almenys el doble que la freqüència màxima del senyal. És a dir, per poder reproduir bé un senyal que s'estigui mesurant, s'haurà de prendre una mostra almenys el doble de vegades que la freqüència màxima d'aquest senyal. En el cas d'aquest projecte, per fer un mostreig precís del ventilador TROTEC TVE-14 la taxa de mostreig per segon haurà de ser com a mínim el doble la freqüència màxima del ventilador. A aquesta taxa de mostreig se li diu Taxa de Nyquist.



Si el mostreig es fa complint la taxa de mostreig que estableix el teorema de Nyquist es podrà evitar un fenomen anomenat aliasing, és a dir, que els senyals se superposin i no es distingeixin l'un de l'altre.

Matemàticament, el teorema de mostreig de Nyquist-Shannon pot formular-se de la següent manera:

Si una funció  $x(t)$  conté no més que una freqüència màxima  $B$  hertz, llavors pot ser completament determinada pels valors discrets presos a intervals d' $1/(2B)$  segons.

En termes de la freqüència de mostreig  $f_s$  (la quantitat de mostres preses per segon), això es tradueix en:

$$f_s \geq 2B$$

On  $B$  és la freqüència màxima en el senyal.

En aquest TFG el senyal que es vol mesurar és la vibració del ventilador. L'aparell fa 1250 RPM quan funciona en la velocitat més alta, és a dir, 1250 revolucions/60 segons = 20,83 revolucions per segon, o hertz (Hz). En funció del mencionat Teorema de Mostreig de Nyquist -Shannon la freqüència de mostreig de la vibració haurà de ser, almenys, el doble, és a dir, mínim  $20,83 \cdot 2 = 41,66$  Hz.

La freqüència de mostreig establerta en els codis d'aquest projecte és de 60 Hz, per tant, la taxa de mostreig és considerablement més alta que l'establerta pel teorema de Nyquist. D'aquesta manera no només s'està complint el teorema que defineix que la freqüència de mostreig ha de ser com a mínim el doble, sinó que també arriba a ser quasi el triple. D'aquesta manera, es compta amb un marge de seguretat que permet mostrejar possibles vibracions inesperadament més altes en el ventilador.

En resum, per poder reconstruir el senyal de vibració del ventilador correctament, complir el Teorema Nyquist-Shannon és indispensable. En el cas que el mostreig estigués per sota de la taxa establerta pel Teorema, es podria produir l'aliasing on el senyal podria ser interpretat erròniament i es perdria informació.

## 5. Model d'AA amb Scikit-learn

### 5.1 Adquisició i emmagatzematge de dades de l'acceleròmetre pel model d'AA

Gràcies a l'adquisició i emmagatzematge de dades de l'acceleròmetre Adafruit ADXL345 s'aconseguirà la informació necessària per als models d'aprenentatge automàtic i profund. Aquest és un pas crític en el procés de manteniment predictiu. Les dades recollides seran analitzades pels models d'aprenentatge i aquests detectaran els diferents estats del funcionament normal (velocitat 0,1,2 i 3) en el ventilador. En aquest apartat es presenten els detalls del procés d'adquisició, processament i emmagatzematge de les dades de l'acceleròmetre per a la seva posterior anàlisi.

#### 5.1.1 Inicialització i configuració de l'acceleròmetre

Per començar amb l'adquisició de dades, el primer que s'ha de fer és inicialitzar i configurar l'acceleròmetre. El codi que s'ha desenvolupat en Python per dur a terme l'adquisició de dades, farà servir llibreries específiques per a interactuar amb l'acceleròmetre ADXL345 i el protocol I2C. Amb l'ús d'aquestes llibreries s'establirà comunicació entre la Raspberry Pi 4 i el sensor, a més també permetran configurar els paràmetres d'aquest (rang de mesurament, freqüència de mostreig...) i llegir els valors d'acceleració en els eixos X, Y i Z.

#### 5.1.2 Descripció del codi de recollida de dades pel model d'AA

Per fer possible d'una manera eficient la recollida de dades de l'acceleròmetre s'ha hagut de desenvolupar un codi en Python, que es pot trobar en l'annex [I]. En aquest apartat, explicaré de manera sintetitzada el funcionament d'aquest codi:

- **Activació de l'acceleròmetre ADXL345:** primer de tot, és essencial establir una connexió amb el protocol I2C entre l'acceleròmetre ADXL345 i la Raspberry Pi, és a dir, entre mestre i esclau. Això s'aconseguirà amb la llibreria *busio*. A continuació, amb la llibreria *adafruit\_adxl345* s'activarà l'acceleròmetre per tal d'obtenir la informació de les acceleracions del ventilador.
- **Configuració de paràmetres de mostreig:** abans de començar amb la lectura s'han d'establir els paràmetres amb els que es vol realitzar el mostreig. En el cas d'aquest

TFG s'ha decidit que el temps de cada mostreig sigui d'un minut, és a dir, es recollirà informació de cada anomalia i/o funcionament normal durant 60 segons. La freqüència serà 60Hz tal com s'ha establert a l'apartat [4.4.2]. És a dir, es prendran 60 mostres per segon durant 60 segons.

- **Demanar a l'usuari la ruta i nom per l'arxiu de sortida:** totes les dades que es recullin de l'acceleròmetre ADXL345 s'han de classificar, registrar i guardar en una base dades, que en aquest cas serà un arxiu CSV. Abans de començar amb el mostreig, el sistema demanarà a l'usuari establir un nom per a l'anomalia i/o funcionament normal que s'està registrant, així com una ruta per guardar el CSV on es reflectiran les dades. Per exemple, en el cas d'aquest projecte, s'ha anomenat un registre del funcionament normal com a “Velocitat\_1”.
- **Funció `collect_data()`:** el sensor ADXL345 mesura l'acceleració en els eixos X,Y i Z del ventilador. La funció encarregada de recollir la informació que envia el sensor és `collect_data()`. Durant 60 segons, és a dir, la durada de mostreig, s'executa la funció repetidament, i així, cada vegada que es recull una lectura de l'acceleròmetre es registra en un arxiu CSV amb el nom del funcionament normal i/o anomalia.
- **Gràfics de Matplotlib en temps real:** a l'hora d'agafar les dades del sensor, és interessant tenir alguna cosa que permeti la visualització en temps real de les lectures, ja que així es podrà detectar una desconexió imprevista o qualsevol anomalia en l'àmbit visual. Per aquest objectiu es fa servir *Matplotlib*. Amb aquests gràfics es podrà veure a temps real els valors de l'acceleròmetre en els eixos X, Y i Z. Utilitzant la funció `update_plots()` s'aniran actualitzant els gràfics cada 300 ms.
- **Iniciar la recol·lecció de dades:** en aquest punt es fa una crida a la funció `collect.data()` perquè comenci a executar-se amb `collection_thread.start()`. Així doncs, a cada iteració, `plt.show()` de *Matplotlib* mostrarà el gràfic amb les dades recollides de *Matplotlib* per tal de mostrar les dades a mesura que es van recol·lectant respectant els temps definits.

- **Desar els gràfics de les lectures:** les imatges (captures de pantalla completes) dels gràfics que es veuen a temps reals durant la recollida de dades poden resultar útils, per exemple per il·lustrar les dades a l'hora de treballar amb elles. Per això és interessant guardar les imatges dels gràfics un cop ha acabat la recollida de dades. Es desarà una imatge de cada gràfic (X,Y i Z) en la mateixa ruta del fitxer CSV on s'han desat les lectures. Per fer tot això, es fa servir la funció *save\_plot()*.

## 5.2 Preprocessament de les dades del model d'AA

Com el seu nom indica, el preprocessament de dades és un tractament que s'ha de fer abans que el model d'AA processi la informació. En el context d'aquest TFG, un cop s'han estret les lectures de les acceleracions del ventilador, per tal de reduir el soroll i que les dades presentin cert ordre, aquestes s'hauran de netejar i organitzar. Això facilitarà la feina dels models d'aprenentatge perquè puguin fer les prediccions més eficaçment.

Abans d'introduir les dades recol·lectades en l'etapa de preprocessament, es procedeix a adjuntar totes les dades de les diferents velocitats en un mateix CSV. Això es farà amb el codi de concatenació que es pot trobar en l'annex [II].

### 5.2.1 Descripció del codi pel preprocessament de les dades del model d'AA

Després de la recol·lecció de dades extreta, com s'explica a l'apartat [5.1], s'ha desenvolupat un codi en Python pel preprocessament, que es pot trobar en l'annex [IV]. En els següents punts s'explica aquest procés:

- **Lectura de l'arxiu CSV:** les dades que es volen preprocessar estan a l'arxiu CSV que s'ha creat durant les lectures de les vibracions del ventilador. El primer pas és obrir aquest CSV i llegir les dades per poder fer aquest preprocessament. Per aquest objectiu faig servir la llibreria *Pandas* que serveix per poder llegir i gestionar les dades dins d'un CSV eficientment.
- **Separació de característiques i etiquetes:** les característiques (features) són variables o paràmetres propis d'allò que s'està estudiant per fer prediccions amb AA o AP (en aquest cas, serien els valors de les acceleracions dels eixos del ventilador). Les etiquetes són les classificacions que el model intentarà predir a partir de les

característiques (per exemple, “Velocitat\_1” seria una etiqueta d'un funcionament normal del ventilador). Seguint l'exemple d'aquest TFG, el model a partir de les característiques de les vibracions predirà si està en “Velocitat\_1”, o en altres funcionaments normals. Fer la separació de característiques i etiquetes és molt important, ja que és el que el model necessita per poder fer les seves prediccions.

- **Escalat de característiques:** certs algorismes poden caure en "l'error" de donar més importància a una característica segons l'escala d'aquesta, això fa que el model pugui estar influenciat incorrectament i influeix directament en la seva eficiència. Per evitar aquest risc, es fa l'escalat de característiques o escalat de dades, que consisteix a limitar el rang d'escala de les característiques, normalment aquest rang és [0,1]. D'aquesta manera, totes les característiques tindran una mateixa importància pel model. Per a dur a terme aquest escalat es fa servir l'objecte *StandardScaler*.
- **Codificació d'etiquetes:** mentre que les persones comprenem millor les dades quan estan assignades a paraules, les màquines entenen millor els números. Els algorismes d'AA o AP comprenen millor els valors numèrics que les paraules, per això és interessant codificar les etiquetes assignant-los un valor numèric. A això se li diu codificació d'etiquetes o label encoding i és un pas molt important dins del preprocessament de dades. Per a aquesta funció es farà servir l'objecte *LabelEncoder* de *sklearn.preprocessing*.
- **Guardar l'objecte LabelEncoder:** com és obvi, per poder descodificar les etiquetes a les quals se'ls hi ha donat un valor numèric, és important guardar l'objecte *LabelEncoder*. Així es podrà assegurar la coherència en la descodificació d'etiquetes numèriques a les seves categories originals durant l'avaluació i les prediccions del model. En guardar l'objecte en un arxiu *pickle* (.pkl), es manté l'assignació d'etiquetes numèriques a categories per al seu futur ús.
- **Guardar l'objecte StandardScaler:** tal com s'ha comentat sobre l'objecte *LabelEncoder*, també és important guardar l'objecte *StandardScaler*, és a dir, l'objecte que s'encarrega de normalitzar l'escala de les dades per evitar prediccions incongruents. L'objecte es guarda en un arxiu *pickle* (.pkl) per a mantenir la

coherència en l'escalat de característiques en noves mostres de dades durant l'avaluació i les prediccions del model. Així, quan s'introdueixen noves característiques en una següent lectura de dades, aquestes seguiran el mateix escalat que la resta.

- **Guardar les dades del preprocessament en un nou arxiu CSV:** un cop s'han dut a terme els diferents passos descrits, el preprocessament es pot donar per finalitzat. Perquè els models rebin i llegeixin aquestes dades preprocessades aquestes s'hauran de guardar en un nou arxiu CSV. D'aquesta manera serà fàcil carregar les dades en futurs passos del projecte, en concret, en l'entrenament i avaluació del model d'AA.

Per resumir, per aconseguir que les dades estiguin endreçades i en el millor format possible per l'aprenentatge automàtic el preprocessament és un pas fonamental. Les tècniques d'escalat i codificació d'etiquetes fan que les dades tinguin una millor qualitat per ferles servir en el model d'aprenentatge automàtic.

### 5.3 Entrenament i avaluació del model d'AA

Un cop finalitzats els processos de recollida i preprocessament de les dades de l'acceleròmetre Adafruit ADXL345, el següent pas és entrenar i avaluar el model d'AA. En aquest apartat, es justificarà la decisió del tipus d'entrenament que s'ha escollit pel model d'AA, així com una descripció detallada del codi desenvolupat tant per a l'entrenament com per l'avaluació.

#### 5.3.1 Selecció i justificació de l'algorisme d'entrenament del model d'AA

De totes les opcions disponibles per a l'entrenament del model d'AA en aquest TFG, s'ha optat per utilitzar l'algorisme de boscos aleatoris (Random Forest). A continuació, es mostren els motius pels quals s'ha seleccionat aquest algorisme per sobre d'altres:

- **Robustesa i versatilitat:** els Random Forest o boscos aleatoris són un tipus d'algorisme que pertany a l'aprenentatge supervisat i estan basats en arbres de decisió. Els arbres de decisió busquen trobar la resposta a una pregunta. Per fer-ho formularan preguntes relacionades (basades en certs criteris), aquestes preguntes s'aniran ramificant en noves preguntes i respostes fins a arribar a una conclusió final. Els

boscors aleatoris combinen un conjunt d'arbres de decisió, el que aporta al model estabilitat i precisió. D'aquesta manera s'obté un model més robust i versàtil que amb un sol arbre de decisió.

- **Capacitat per a manejar característiques sorolloses:** atès que en aquest projecte es recullen les dades a partir d'un acceleròmetre col·locat en un ventilador per tal de mesurar les vibracions d'aquest, és comprensible que algunes de les lectures tinguin soroll a causa de les condicions en el moment de la presa de dades. Per exemple, si el ventilador està al costat d'una rentadora en funcionament, les lectures es podrien veure afectades per les vibracions d'aquesta. En una empresa també es pot donar que la màquina a la qual es vol fer el manteniment predictiu estigui al costat d'una altra que pugui aportar soroll a les dades. Per això, l'algorisme de boscors aleatoris és una bona opció, ja que aquest té una capacitat per a manejar característiques sorolloses i reduir el sobreajust.
- **Importància de les característiques:** un dels grans avantatges que brinden els Boscors Aleatoris és la seva capacitat per a assenyalar quant d'important és cada característica de les nostres dades. Aquest detall és clau quan es vol classificar funcionaments o anomalies, perquè ajuda a entendre quines són les característiques que realment importen en aquest procés. En conèixer aquestes peces clau, es pot millorar l'eficiència del model. A més, aquesta informació dona la possibilitat de netejar les dades, és a dir, de reduir la seva dimensionalitat. Això es tradueix a treure de l'equació aquelles característiques que no són tan rellevants, la qual cosa permet optimitzar el funcionament del model.
- **Fàcil implementació i ajust:** el codi que s'utilitza per a l'entrenament i avaluació del model, fa servir la llibreria *sklearn.ensemble* per implementar l'algorisme de boscors aleatoris. Un gran avantatge que té aquest mètode és que és molt senzill d'implementar i ajustar els hiperparàmetres del model. Un dels hiperparàmetres és, per exemple, el nombre d'arbres que hi haurà en aquest bosc aleatori, ja que si són massa pocs, el model podria ser poc precís i si són molts, massa complex. Finalment, en aquest TFG, s'ha seleccionat un valor de 100 arbres, que és un valor equilibrat.

En resum, l'algorisme de boscos aleatoris s'ha seleccionat per a aquest treball a causa de la seva robustesa, versatilitat, capacitat per a manejar característiques sorolloses i la seva senzilla implementació. En l'avaluació del model, s'acabarà de concloure si realment aquest algorisme s'ajusta correctament per a la detecció i classificació de les diferents classes de funcionament normal en el ventilador.

### 5.3.2 Descripció del codi per l'entrenament del model d'AA

El model d'AA que s'ha desenvolupat en aquest TFG ha de ser capaç de detectar els diferents funcionaments normals del ventilador (les velocitats). Per tal de complir amb aquest objectiu, s'ha desenvolupat un codi en Python basat en la llibreria *Scikit-learn*. A l'apartat [3.6.3] s'ha mencionat que aquesta llibreria és una eina molt coneguda en l'entorn de l'aprenentatge automàtic, ja que es fa servir amb abundància tant per l'entrenament, l'avaluació i optimització dels models.

En els següents punts es pot observar el pas a pas del codi desenvolupat (que es pot trobar en l'annex [V]) pel procés d'entrenament i avaluació del model:

- **Importar les biblioteques necessàries:** per tal de que el model funcioni correctament s'han d'importar les següents biblioteques:
  - **numpy:** una biblioteca per a treballar amb matrius numèriques i realitzar operacions matemàtiques.
  - **sklearn.model\_selection:** conté la funció *train\_test\_split* per a dividir les dades en conjunts d'entrenament i prova.
  - **sklearn.ensemble:** conté la implementació de l'algorisme *RandomForestClassifier*.
  - **sklearn.metrics:** proporciona funcions per a calcular mètriques de rendiment com *classification\_report* i *confusion\_matrix*.



- **pickle:** una biblioteca per a guardar i carregar objectes de Python en un arxiu binari.
- **Definir la funció 'train\_and\_save\_model':** aquesta funció accepta dos arguments: la ruta de l'arxiu CSV que conté les dades preprocessades i la ruta on es guardarà el model entrenat. La funció realitza els següents passos:
  - **Llegir les dades del preprocessament:** primer de tot, s'ha de carregar l'arxiu CSV amb les dades preprocessades (el preprocessament s'ha comentat en l'apartat [5.2]). Per això, es fa servir la funció *np.loadtxt*, de *Numpy*. Tal com s'ha comentat a l'apartat [5.2.1], els models d'aprenentatge interpreten millor els valors numèrics. La funció mencionada converteix la informació en una matriu numèrica.
  - **Separar les característiques i les etiquetes:** la matriu de dades es divideix en dues parts. La variable 'X' conté les característiques (totes les columnes menys la última) i la variable 'Y' conté les etiquetes de les classes (última columna). Aquestes característiques i etiquetes s'utilitzaran per a entrenar i avaluar el model.
  - **Dividir les dades en conjunts d'entrenament i prova:** per tenir una noció de l'evolució del model que s'està entrenant, és interessant dividir les dades de manera que una part de les dades serveixi per entrenar el model i la resta es faci servir per avaluar si el seu rendiment és correcte. És a dir, no podem saber si el codi que estem desenvolupant està millorant si no l'avaluem, i aquesta avaluació es fa amb una part de les dades que introduïm. En l'entorn de la intel·ligència artificial es recomana que la proporció sigui 80-20, el 80% de les dades serà per l'entrenament i el 20% per avaluar aquest. Això es farà amb la funció *train\_test\_split* de *Scikit-learn*.
  - **Entrenar el model:** es crea un objecte *RandomForestClassifier* amb 100 arbres de decisió i una llavor aleatòria fixa (*random\_state=42*) per a garantir la

reproductibilitat dels resultats. El model s'entrena utilitzant el mètode *fit* i les dades d'entrenament ( $X_{train}$ ,  $Y_{train}$ ).

- **Avaluar el model:** tal com s'ha comentat en el pas de divisió de dades, les dades introduïdes en el model s'han separat en dos grups, el 80% per a l'entrenament del model i el 20% restant per a l'avaluació d'aquest. D'aquesta manera, el model haurà de predir les classes (Velocitat 0, 1, 2 i 3) amb el 20% de les dades ( $X_{test}$ ) fent servir el mètode *predict*.

Un cop fet això, s'han de calcular les mètriques de rendiment, és a dir, els paràmetres que es fan servir per avaluar l'eficàcia d'un model d'AA. Aquestes mètriques són, tal com explica l'apartat [3.7.1], la precisió, l'exhaustivitat, el F1-score i la matriu de confusió. La funció *classification\_report* em donarà un resum d'aquestes mètriques, així s'avaluarà si el model classifica de manera adequada els funcionaments normals del ventilador. Per tenir una representació més visual, la funció *confusion\_matrix* exposa visualment les prediccions correctes i incorrectes.

- **Guardar el model:** per no haver d'entrenar el model cada vegada que es vulgui utilitzar per a detectar els estats del funcionament normal del ventilador, s'haurà de guardar. Per guardar el model, es fa servir la llibreria *pickle*, que guardarà l'objecte del classificador en un arxiu binari en la ruta que s'indiqui. Aquesta ruta s'ha d'escriure en *model\_path*. D'aquesta manera, no serà necessari tornar a reentrenar el model per a quan es vulgui utilitzar de nou. Així doncs, només farà falta carregar l'arxiu.
- **Executar la funció 'train\_and\_save\_model':** un cop definida la funció *train\_and\_save\_model* (que és la funció que entrenarà i avaluarà el model), es pot procedir a l'execució d'aquesta. Abans d'això, només quedarà indicar la ruta de l'arxiu CSV on es torben les dades preprocessades que es volen fer servir per a l'entrenament i l'avaluació i indicar la ruta on es vol guardar el model un cop el procés es finalitzi. Totes dues rutes es passen com a arguments en la funció *train\_and\_save\_model*. En executar la funció *train\_and\_save\_model*, és dura a terme el procés d'entrenament i avaluació del model.

Seguint els passos previament descrits s'aconsegueix crea un model d'AA. Aquest model pot ser posteriorment utilitzat per a implementar el sistema de manteniment predictiu.

### 5.3.3 Avaluació del model d'AA:

Tal com s'ha explicat en l'apartat [5.3.2], quan s'entrena el model d'aprenentatge automàtic es reserva una part de les dades per avaluar el resultat d'aquest entrenament. En el cas d'aquest TFG, i seguint la tendència en l'entorn de la IA, s'ha reservat un 20% de les dades per la fase d'avaluació del model.

Aquesta avaluació es representa a través d'unes mètriques (la precisió, l'exhaustivitat, el F1-score i la matriu de confusió). A continuació es poden veure fórmules matemàtiques, càlculs i els resultats d'aquestes mètriques en el model d'AA que s'ha utilitzat per predir classes del funcionament normal del ventilador.

Tal com s'ha mencionat en la descripció del codi per l'entrenament del model d'AA, la funció *classification\_report* mostra un resum de les mètriques d'avaluació i la funció *confusion\_matrix* presenta una matriu amb totes les prediccions correctes i incorrectes. Aquest informe de classificació i matriu de confusió es poden veure en les taules [3] i [4] correlativament.

	<b>precision</b>	<b>recall</b>	<b>f1-score</b>	<b>support</b>
0.0	0.96	0.98	0.97	646
1.0	0.44	0.48	0.46	623
2.0	0.45	0.40	0.42	637
3.0	0.54	0.53	0.53	584
<b>accuracy</b>			0.60	2490
<b>macro avg</b>	0.60	0.60	0.60	2490
<b>weighted avg</b>	0.60	0.60	0.60	2490

**Taula - 3** Informe de classificació del model d'AA

```
[[633 5 6 2]
 [ 13 299 192 119]
 [ 9 227 257 144]
 [ 6 146 122 310]]
```

**Taula - 4** Matriu de confusió del model d'AA

- **Matriu de Confusió:** aquesta matriu és una taula on es veuran totes les prediccions per cada classe del model, correctes i incorrectes. D'aquesta manera, la taula compara les prediccions i les etiquetes reals. Per entendre la matriu, horitzontalment, fila a fila, es poden observar les etiquetes reals de cada classe. En aquest cas, la primera fila són les instàncies de la Velocitat 0 reals, la segona fila pertany a la Velocitat 1, la tercera fila a la Velocitat 2 i la quarta a la Velocitat 3.

Si es llegeix la matriu verticalment, s'obtidran les prediccions que ha fet el model. La primera columna són les prediccions on el model ha detectat la Velocitat 0 del ventilador, la segona columna les prediccions de la Velocitat 1, la tercera la Velocitat 2 i la quarta són les prediccions de la Velocitat 3.

Es pot veure que el model ha detectat 633 instàncies de la Velocitat 0 correctament, ja que estan a la columna i fila de la Velocitat 0. No obstant això, en 5 instàncies que eren en realitat Velocitat 0 (columna 2, fila 1), el model ha predit que es tractava de la Velocitat 1.

En l'avaluació del model, la matriu de confusió és la següent:

```
[[633, 5, 6, 2],
 [ 13, 299, 192, 119],
 [ 9, 227, 257, 144],
 [ 6, 146, 122, 310]]
```

Els elements de la diagonal principal representen les prediccions correctes (TP) per a cada classe:

$$TP (\text{Classe } 0) = 633$$

$$TP (\text{Classe } 1) = 299$$

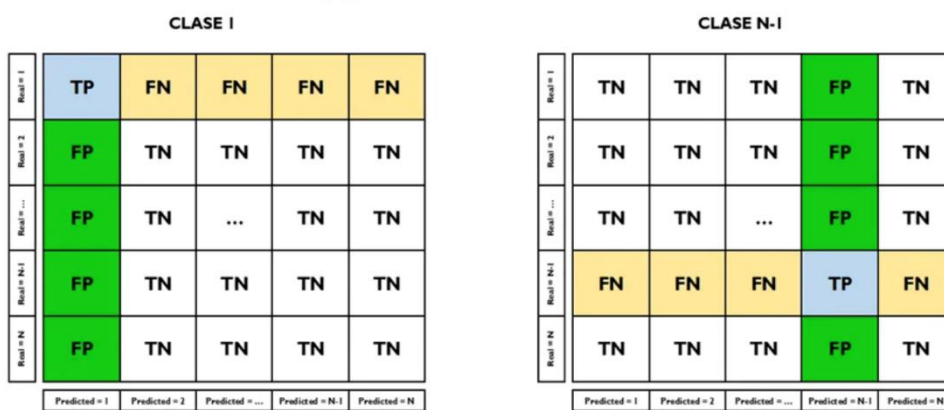
$$TP (\text{Classe } 2) = 257$$

$$TP (\text{Classe } 3) = 310$$

Tal com s'ha explicat, els números fora de la diagonal principal són prediccions incorrectes. Per exemple el cas del número 5 (posicionat en la segona posició de la fila 1) és en realitat una Velocitat 0 etiquetada pel model com a Velocitat 1. En aquest cas, és un fals negatiu (FN). Això passarà amb tota la primera fila fora de la diagonal principal, són falsos negatius de la Velocitat 0. Continuant amb aquesta lògica, els valors de la columna 1 fora de la diagonal principal (els números 13, 9 i 6) són falsos positius per a la Classe Velocitat 0.

$$FN (\text{Classe } 0) = 5 (\text{Classe } 1) + 6 (\text{Classe } 2) + 2 (\text{Classe } 3) = 13$$

$$FP (\text{Classe } 0) = 13 (\text{Classe } 1) + 9 (\text{Classe } 2) + 6 (\text{Classe } 3) = 28$$



**Figura - 9** Matriu de Confusió per a un classificador multiclasse. Referència bibliografia: [46]

La matriu de confusió és molt útil per visualitzar els resultats del model en cada classe i identificar problemes com la possible desigualtat entre classes.

- **Exactitud (Accuracy):** L'exactitud calcula la proporció de prediccions correctes entre totes les prediccions realitzades pel model. Tal com ja s'ha remarcat a l'apartat [3.7.1], en el cas de classificacions multiclasse, com és el d'aquest TFG (ja que el model ha de predir les diferents velocitats del funcionament normal del ventilador: 0,1,2 i 3), per calcular l'exactitud s'han de dividir les prediccions correctes entre el total de prediccions.

$$Exactitud = (\Sigma TP) / (\Sigma TP + \Sigma FP + \Sigma FN)$$

Per el resultats del model, l'exactitud es calcularia com:

$$Exactitud = (633 + 299 + 257 + 310) / 2490 \approx 0.60$$

- **Precisió:** tal com s'ha explicat en la secció de fonaments teòrics a l'apartat [3.7.1], la precisió és la proporció dels resultats positius correctes (TP) entre totes les prediccions positives que ha fet el model (la suma entre els veritables positius i els falsos positius).

$$Precisió = TP / (TP + FP)$$

A continuació es presenta la precisió per cada classe o Velocitat:

$$\begin{aligned} Precisió (Classe 0) &= TP (Classe 0) / (TP (Classe 0) + FP (Classe 0)) \\ &= 633 / (633 + 13 + 9 + 6) = 633 / 661 \approx 0.96 \end{aligned}$$

$$\begin{aligned} Precisió (Classe 1) &= TP (Classe 1) / (TP (Classe 1) + FP (Classe 1)) \\ &= 299 / (299 + 5 + 227 + 146) = 299 / 677 \approx 0.44 \end{aligned}$$

$$\begin{aligned} \text{Precisió (Classe 2)} &= TP (\text{Classe 2}) / (TP (\text{Classe 2}) + FP (\text{Classe 2})) \\ &= 257 / (257 + 6 + 192 + 122) = 257 / 577 \approx 0.45 \end{aligned}$$

$$\begin{aligned} \text{Precisió (Classe 3)} &= TP (\text{Classe 3}) / (TP (\text{Classe 3}) + FP (\text{Classe 3})) \\ &= 310 / (310 + 2 + 119 + 144) = 310 / 575 \approx 0.54 \end{aligned}$$

- **Recall (Sensibilitat):** el Recall és la proporció de casos positius identificats. Es calcula dividint les prediccions positives reals (TP) entre totes les etiquetes que són positives en la realitat (TP+FN).

$$\text{Recall} = TP / (TP + FN)$$

A continuació es mostren els càlculs pel Recall en el cas concret d'aquest TFG:

$$\begin{aligned} \text{Recall (Classe 0)} &= TP (\text{Classe 0}) / (TP (\text{Classe 0}) + FN (\text{Classe 0})) \\ &= 633 / (633 + 5 + 6 + 2) = 633 / 646 \approx 0.98 \end{aligned}$$

$$\begin{aligned} \text{Recall (Classe 1)} &= TP (\text{Classe 1}) / (TP (\text{Classe 1}) + FN (\text{Classe 1})) \\ &= 299 / (299 + 13 + 192 + 119) = 299 / 623 \approx 0.48 \end{aligned}$$

$$\begin{aligned} \text{Recall (Classe 2)} &= TP (\text{Classe 2}) / (TP (\text{Classe 2}) + FN (\text{Classe 2})) \\ &= 257 / (257 + 9 + 227 + 14) = 257 / 637 \approx 0.40 \end{aligned}$$

$$\begin{aligned} \text{Recall (Classe 3)} &= TP (\text{Classe 3}) / (TP (\text{Classe 3}) + FN (\text{Classe 3})) \\ &= 310 / (310 + 6 + 146 + 122) = 310 / 584 \approx 0.53 \end{aligned}$$

- **F1-score:** la puntuació F1 es calcula com la mitjana harmònica entre la precisió i el recall:

$$F1 - score = 2 \cdot \frac{\text{Precisió} \cdot \text{Recall}}{\text{Precisió} + \text{Recall}}$$

Llavors, l'F1-score de cada classe en aquest cas seria:

$$F1 - score (Classe 0) = 2 \cdot \frac{0.96 \cdot 0.98}{0.96 + 0.98} \approx 0.97$$

$$F1 - score (Classe 1) = 2 \cdot \frac{0.44 \cdot 0.48}{0.44 + 0.48} \approx 0.46$$

$$F1 - score (Classe 2) = 2 \cdot \frac{0.45 \cdot 0.40}{0.45 + 0.40} \approx 0.42$$

$$F1 - score (Classe 3) = 2 \cdot \frac{0.54 \cdot 0.53}{0.54 + 0.53} \approx 0.53$$

- **Support:** Tal com s'ha explicat a l'apartat [3.7.1], el suport és simplement el nombre d'instàncies reals de cada classe en el conjunt de dades de prova. En l'avaluació reflectida al principi d'aquest apartat, el suport per a cada classe es mostra en la columna "support" de l'informe de classificació:

Suport (Classe 0) = 646

Suport (Classe 1) = 623

Suport (Classe 2) = 637

Suport (Classe 3) = 584

- **Mitjana Macro i Ponderada:** Tal com s'ha explicat a l'apartat [3.7.1], les mitjanes Macro (macro avg) i Ponderada (weighted avg) són mitjanes que es fan de les mètriques de cada classe (precisió, recall i F1-score). La Mitjana Ponderada té en compte la possible desigualtat de les etiquetes, ponderant segons el suport. Al contrari, la Mitjana Macro fa la mitjana sense ponderar. En aquest cas són molt similars perquè el nombre d'instàncies de cada classe són també similars.

Les mitjanes Macro i Ponderada per aquest model, es calculen de la següent manera:

- **Mitjana Macro:**



$$\text{Mitjana Macro Precisió} = \frac{\sum \text{Precisió (Clase } n)}{4} = \frac{0.96 + 0.44 + 0.45 + 0.54}{4} \approx 0.60$$

$$\text{Mitjana Macro Recall} = \frac{\sum \text{Recall (Clase } n)}{4} = \frac{0.98 + 0.48 + 0.40 + 0.53}{4} \approx 0.60$$

$$\text{Mitjana Macro F1 - Score} = \frac{\sum \text{F1 - Score F1 (Clase } n)}{4} = \frac{0.97 + 0.46 + 0.42 + 0.53}{4} \approx 0.60$$

○ **Mitjana Ponderada:**

$$\begin{aligned} \text{Mitjana Ponderada Precisió} &= \frac{\sum \text{Presició (Classe } n) \cdot \text{suport (Classe } n)}{\text{suport total}} = \\ &= \frac{0.96 \cdot 646 + 0.44 \cdot 623 + 0.45 \cdot 637 + 0.54 \cdot 584}{646 + 623 + 637 + 584} \approx 0.60 \end{aligned}$$

$$\begin{aligned} \text{Mitjana Ponderada Recall} &= \frac{\sum \text{Recall (Classe } n) \cdot \text{suport (Classe } n)}{\text{suport total}} = \\ &= \frac{0.98 \cdot 646 + 0.48 \cdot 623 + 0.40 \cdot 637 + 0.53 \cdot 584}{646 + 623 + 637 + 584} \approx 0.60 \end{aligned}$$

$$\begin{aligned} \text{Mitjana Ponderada F1 - Score} &= \frac{\sum \text{F1 - Score (Classe } n) \cdot \text{suport (Classe } n)}{\text{suport total}} = \\ &= \frac{0.97 \cdot 646 + 0.46 \cdot 623 + 0.42 \cdot 637 + 0.53 \cdot 584}{646 + 623 + 637 + 584} \approx 0.60 \end{aligned}$$

Després de calcular totes les mètriques mencionades, els resultats de l'avaluació del model d'aprenentatge automàtic es resumeixen en els següents punts:

- L'exactitud del model en general és d'un 60%, això vol dir que de totes les prediccions que ha fet el model, 60 de cada 100 són correctes.

- Les mètriques classe per classe, precisió, Recall i F1-Score són diferents, sobretot amb gran diferència en la detecció de la Velocitat 0 respecte a les altres velocitats. El model detecta la Classe 0, és a dir, la Velocitat 0, amb una precisió del 96%, un Recall de 98% i una F1-Score de 97%. La detecció de la resta de velocitats té mètriques menys altes, que varien entre el 40% i el 54%.
- A la Matriu de Confusió es pot veure el que ja he comentat en el punt anterior. La Classe 0 és detectada amb resultats bons pel model, però hi han més dificultats per predir la resta de classes. La quantitat de prediccions errònies del model en les velocitats 1, 2 i 3 són nombroses.

Es pot concloure que el model té un bon rendiment en la Classe 0, però té dificultats per a classificar correctament les Classes 1, 2 i 3. L'exactitud global del model és del 60%, la qual cosa pot ser insuficient en funció de les expectatives i els requisits del projecte. Per a millorar el rendiment del model, es podrien explorar diferents tècniques de preprocessament, ajust d'hiperparàmetres o fins i tot provar altres algorismes de classificació.

## 5.4 Desplegament del model d'aprenentatge automàtic

Fins aquest moment, s'ha implementat un sistema per agafar les dades del ventilador, s'han preprocessat les dades i s'ha entrenat el model d'AA. Per tal de poder detectar les diferents velocitats del funcionament normal del ventilador en temps real, fa falta desenvolupar un deployment. El deployment, en català desplegament, es refereix al procés d'aplicació del model entrenat per a detectar en temps real les classes. En aquest apartat s'explicarà en detall el procés que segueix el deployment per tal de complir amb la seva funció.

### 5.4.1 Descripció del codi per al desplegament del model d'aprenentatge automàtic

Pel desplegament del model d'AA, igual que en els passos previs, s'ha desenvolupat un codi en Python, que es pot trobar a l'annex [VI]. A continuació es veurà pas a pas el funcionament d'aquest codi per a la implementació i desplegament del model que ha de detectar i classificar les velocitats del funcionament normal del ventilador TROTEC TVE 14 en temps real.

- **Inicialització de l'acceleròmetre:** en primer lloc, s'han d'importar les llibreries que es faran servir: *os*, *time*, *board*, *busio*, *adafruit\_adxl34x*, *StandardScaler*, *LabelEncoder*, *pickle* i *numpy*. Per tal d'aconseguir la comunicació entre la Raspberry Pi 4 i l'acceleròmetre ADXL345 es fan servir les següents funcions: Amb la funció *busio.I2C(board.SCL, board.SDA)* es procedeix a iniciar la comunicació I2C. Finalment, la funció *adafruit\_adxl34x.ADXL345(i2c)* es fa servir per crear una instància de l'acceleròmetre ADXL345.
- **Carregar el model entrenat:** el model d'AA, que s'ha entrenat i avaluat prèviament, està guardat en un arxiu *.pkl* anomenat *model\_entrenat.pkl*. Aquest es carregarà amb la funció *pickle.load()* per fer-lo servir en aquest desplegament. Tal com s'ha explicat a l'apartat [5.3.1], aquest és un model que fa servir boscos aleatoris amb l'objectiu de predir en temps real les velocitats del funcionament normal del ventilador a partir de les dades de l'acceleròmetre connectat a aquest.
- **Carregar el StandardScaler i LabelEncoder:** a més de carregar el model entrenat, també es carreguen el *StandardScaler* i el *LabelEncoder* des dels seus respectius arxius (*scaler.pkl* i *label\_encoder.pkl*) utilitzant *pickle.load()*. El *StandardScaler* es fa servir per a escalar les dades de l'acceleròmetre abans d'ingressar-los al model, mentre que el *LabelEncoder* es fa ús per a descodificar les etiquetes de les classes després de realitzar les prediccions.
- **Definir la taxa de mostreig i el retard entre mesuraments:** la taxa o freqüència de mostreig, tal com s'ha explicat a l'apartat [4.4], indica quantes mesures de les acceleracions del ventilador fa el sensor en un segon. En aquest TFG, tal com s'ha aclarit a l'apartat [4.4.2], s'ha establert que aquesta taxa sigui de 60 Hz, és a dir, 60 recol·leccions de les dades cada segon. Per una altra banda, el retard entre mesuraments, al contrari, es calcula amb la inversa, (1/60), aquest serà el temps que el programa en Python espera abans de fer la següent lectura.
- **Detecció de les classes (velocitats) en temps real:** arriba el moment clau, el codi es posarà en marxa per començar a fer les prediccions del model en temps real i detectar les diferents velocitats del ventilador mentre aquest està en funcionament. Per aquest

motiu, el codi entrarà en bucle (es repetirà constantment), efectuant els següents passos en cada repetició:

- Les dades que processarà el model s'extreuen de l'acceleròmetre col·locat al ventilador. Aquestes seran les dades que s'han de llegir en temps real per tal que el model detecti les diferents velocitats del funcionament normal del ventilador. Les dades es recullen del sensor amb la funció *accelerometer.acceleration*. A més, per tal que les dades siguin d'un format adequat pel model, s'emmagatzemen en un array de *Numpy* i es redimensionen.
- Les dades extretes de l'acceleròmetre s'han de preprocessar, per tal que estiguin organitzades i siguin de fàcil lectura pel model. Aquest preprocessament es fa amb el *StandardScaler* que s'haurà carregat abans i que s'ha generat en el moment del preprocessament de les dades explicat en l'apartat [5.2]. Part del preprocessament de les dades és l'escalat de característiques, això es fa amb la funció *scaler.transform()*.
- Un cop el model comença a rebre les dades preprocessades, el següent pas és començar amb les prediccions, és a dir, el model d'AA ha de detectar les diferents velocitats del funcionament normal del ventilador. Això ho fa gràcies a l'entrenament previ que ha rebut (resumit en l'apartat [5.3]). El que fa el model és predir l'etiqueta de velocitat del ventilador segons les dades que rep. Aquest últim pas ho fa amb la funció *model.predict()*.
- Tal com s'ha assenyalat a l'apartat [5.2.1], les màquines (en aquest cas el model d'aprenentatge artificial) processen millor els valors numèrics. En el preprocessament de dades del model s'ha fet una codificació de les etiquetes (les diferents classes de velocitats del ventilador). Aquestes ara tindran un codi numèric. En aquest pas del deployment, es descodifiquen les etiquetes utilitzant el *LabelEncoder* que s'ha carregat prèviament en el programa. En concret, s'encarrega la funció *label\_encoder.inverse\_transform()*, que converteix el valor numèric de l'etiqueta en el nom corresponent d'aquesta, que en aquest cas serà una de les velocitats del ventilador (0, 1, 2 o 3). D'aquesta

manera en el desplegament, quan el model predigui les classes a temps real, es veurà cada etiqueta pel seu nom, i no pel seu codi numèric.

- S'imprimeix el nom de l'anomalia detectada en la consola o terminal utilitzant la funció `print()`. D'aquesta manera, es proporciona una retroalimentació en temps real sobre les classes (velocitats) detectades pel sistema.
- Prèviament, s'ha explicat que el retard entre mesures és el temps que el programa en Python espera abans de fer la següent lectura. El programa es pausa durant el temps establert pel retard per evitar la saturació per excés de dades. Això es fa amb la funció `time.sleep()`. El programa s'atura durant el temps que marca el retard, i així s'aconsegueix que només es recullin les dades del sensor amb la freqüència establerta per la taxa de mostreig.

En resum, el desplegament del model és un codi que obté les dades de l'acceleròmetre en temps real, les preprocessa, i passa aquestes dades pel model prèviament entrenat per tal de poder detectar les velocitats del funcionament normal del ventilador en temps real. La velocitat detectada per aquest codi serà impresa en el terminal. Aquest procés s'anirà repetint fins que s'aturi el programa manualment. D'aquesta manera podria estar dies, setmanes o mesos monitorant el funcionament del ventilador.

## 6. Model d'AP amb Edge Impulse

Després d'observar els resultats obtinguts amb el model d'AA, s'ha decidit prova amb una altra alternativa d'IA per veure si es pot aconseguir un millor rendiment. S'ha optat per provar amb un model d'aprenentatge profund, ja que l'algorisme de classificació en aquests són diferents de l'utilitzat al model d'AA. Pel desenvolupament del model d'AP s'ha decidit fer servir la plataforma Edge Impulse, una pàgina web que facilita l'entrenament de models d'IA amb un entorn user-friendly. Aquest a més permet de manera senzilla jugar amb els paràmetres del model per tal d'afinar-lo i assolir el millor rendiment possible sense necessitat de canviar tot el codi de programació com és el cas del model d'AA.

### 6.1 Adquisició i emmagatzematge de dades per el model d'AP amb Edge Impulse

#### 6.1.1 Connexió de la Raspberry Pi a la plataforma Edge Impulse

Primerament, la Raspberry Pi 4 s'ha de connectar amb la plataforma Edge Impulse per tal que les dades de l'acceleròmetre puguin ser enviades a aquesta. El primer pas que s'ha de realitzar és connectar de manera física la Raspberry Pi a l'acceleròmetre Adafruit ADXL345. Es pot trobar l'explicació de la connexió física a l'apartat [4.2] d'aquest TFG.

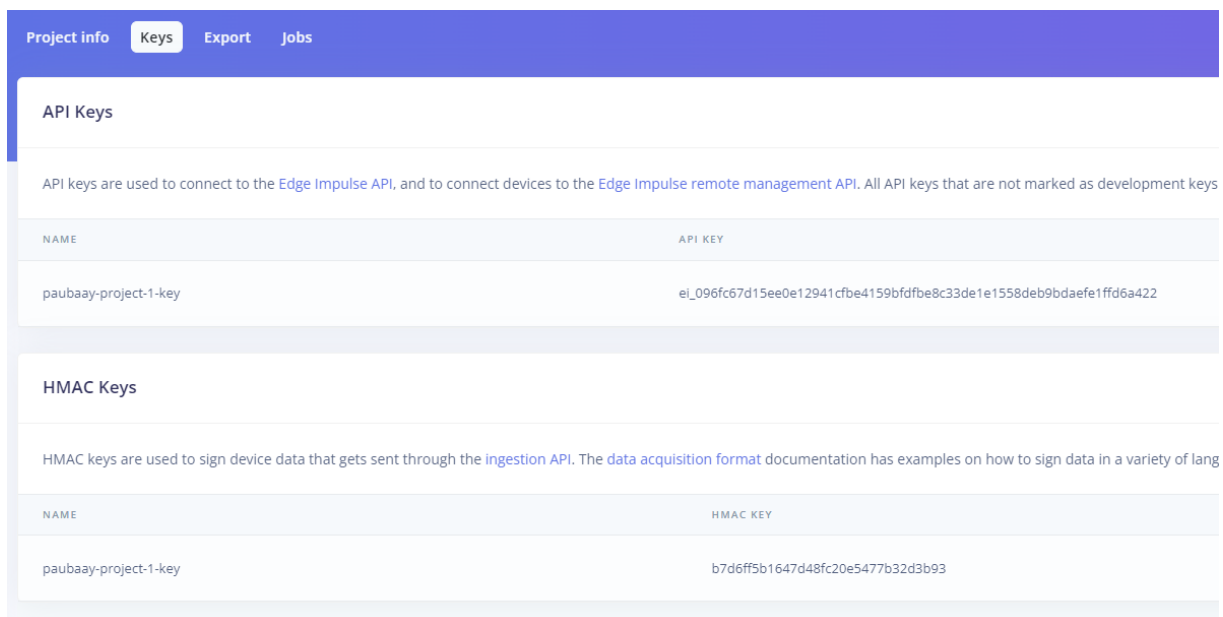
Un cop ja està la connexió física establerta entre la Raspberry Pi i l'acceleròmetre, el següent pas que s'ha de dur a terme és la connexió de la Raspberry Pi a la plataforma Edge Impulse. Per tal d'assolir aquesta connexió, és important tenir dintre de la Raspberry Pi les dependències o llibreries necessàries instal·lades, per tal que posteriorment, l'execució del codi de recollida de dades de l'acceleròmetre a la plataforma Edge Impulse sigui exitosa.

Seguint el manual que es pot trobar en la mateixa web oficial de la plataforma Edge Impulse, amb referència en aquest TFG [22], he instal·lat les dependències en el terminal de la Raspberry Pi amb el codi que es proporciona en l'annex [VII].

El següent punt clau per tal de poder enviar les dades a la plataforma és agafar la HMAC KEY i l'API KEY associades al compte dintre de la pàgina web d'Edge Impulse. Tal com es mostra a la figura [10], a la pestanya "Keys" es poden trobar aquestes claus.

A continuació es fa una petita descripció sobre que són les claus HMAC i API:

- **HMAC KEY:** la clau HMAC (Hash-based Message Authentication Code) no deixa de ser un tipus de credencial o firma digital que serveix per autoritzar sol·licituds a plataformes, sobretot en entorns de plataformes en el núvol com és Edge Impulse.
- **API KEY:** la clau API (Application Programming Interfaces) és un identificador únic per autenticar la connexió entre dos programes computacionals, en aquest cas entre el codi en Python de la Raspberry Pi i la plataforma Edge Impulse.



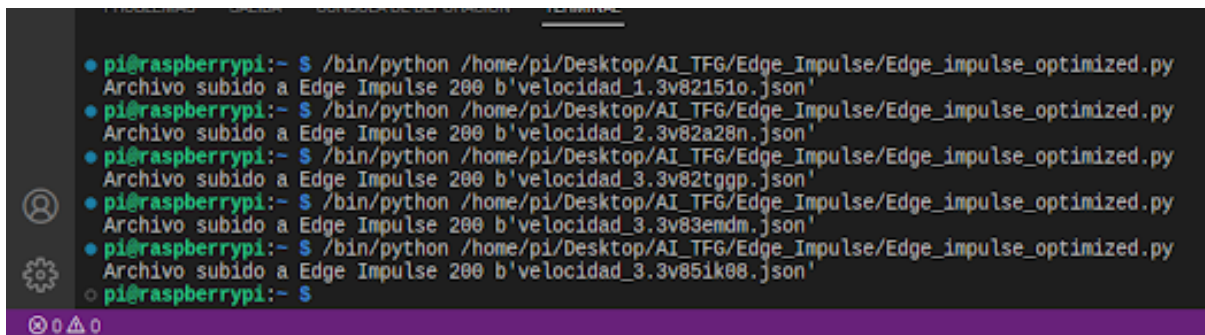
**Figura - 10** Captura de pantalla de la pestanya “Keys” de Edge Impulse per a l’obtenció de les claus API i HMAC

Un cop ja estan totes les dependències instal·lades, per tal de comprovar si la recollida de les dades és exitosa en la plataforma Edge Impulse, s'executarà l'arxiu Python, que es pot trobar en l'annex [VIII], per tal d'enviar les dades al compte d'Edge Impulse.

No obstant això, abans d'executar-lo s'han de definir les claus HMAC i API dins del codi. També s'haurà de canviar el valor de *x-file-name* cada cop que es vulgui enregistrar una etiqueta o classe nova. Així, cada classe en la qual s'està recollint les dades tindrà un nom diferent. A continuació es mostren els camps a modificar:

- **HMAC\_KEY:** Tal com he explicat anteriorment, aquesta constant s'obté de la pàgina web d'Edge Impulse i és personal per a cada usuari.
- **API\_KEY:** Aquesta constant també es pot trobar a la plataforma d'Edge Impulse i també és personal per a cada usuari.
- **x-file-name:** Aquesta variable és la variable que s'ha de canviar per tal d'introduir el nom de l'etiqueta per la qual volem enregistrar les dades.

Un cop ha finalitzat l'execució del codi, apareix en el terminal el següent missatge confirmant que la recollida de dades ha tingut èxit: “Archivo subido a Edge Impulse 200 b 'velocidad\_X\_3v8151o.json' “. En la següent figura [11], es pot visualitzar el missatge de sortida del terminal.

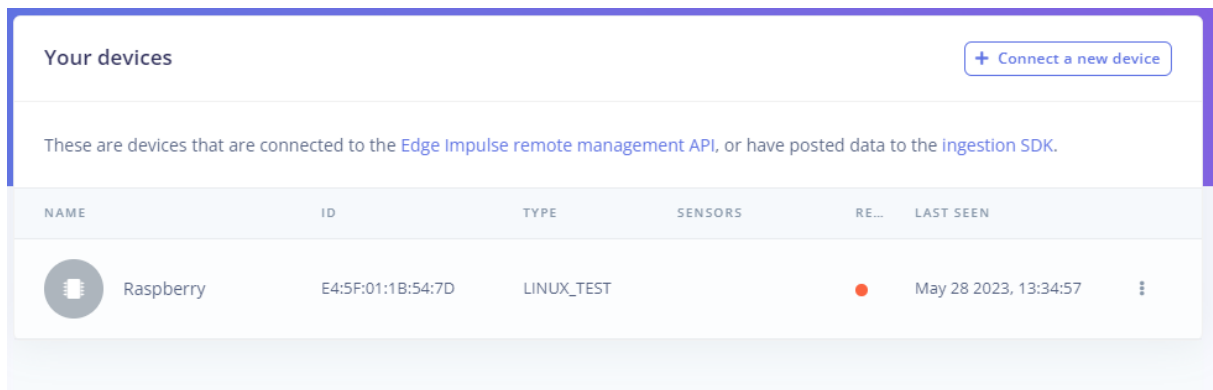


```
pi@raspberrypi:~$ /bin/python /home/pi/Desktop/AI_TFG/Edge_Impulse/Edge_impulse_optimized.py
Archivo subido a Edge Impulse 200 b'velocidad_1.3v82151o.json'
pi@raspberrypi:~$ /bin/python /home/pi/Desktop/AI_TFG/Edge_Impulse/Edge_impulse_optimized.py
Archivo subido a Edge Impulse 200 b'velocidad_2.3v82a28n.json'
pi@raspberrypi:~$ /bin/python /home/pi/Desktop/AI_TFG/Edge_Impulse/Edge_impulse_optimized.py
Archivo subido a Edge Impulse 200 b'velocidad_3.3v82tggp.json'
pi@raspberrypi:~$ /bin/python /home/pi/Desktop/AI_TFG/Edge_Impulse/Edge_impulse_optimized.py
Archivo subido a Edge Impulse 200 b'velocidad_3.3v83emdm.json'
pi@raspberrypi:~$ /bin/python /home/pi/Desktop/AI_TFG/Edge_Impulse/Edge_impulse_optimized.py
Archivo subido a Edge Impulse 200 b'velocidad_3.3v851k08.json'
pi@raspberrypi:~$
```

**Figura - 11** Captura de pantalla missatge sortida terminal codi Python de recollida de dades per a Edge Impulse

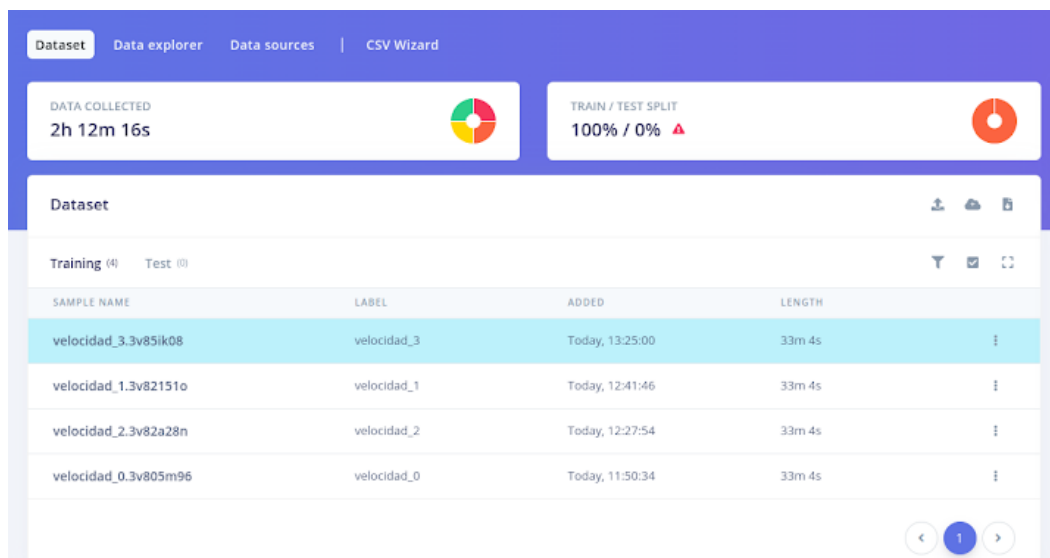
Un cop executat el codi per primer cop i havent imprès el missatge “Archivo subido a Edge Impulse 200 b 'velocidad\_X\_3v8151o.json' “. s'ha de comprovar dintre de la plataforma Edge Impulse que realment l'operació hagi tingut èxit. Per tal de comprovar-ho, s'ha d'anar a la pestanya "Your devices", on ha d'aparèixer, tal com es mostra en la següent figura [12], la Raspberry Pi en el llistat d'equips. Si tot és correcte, a la columna “last seen” haurà d'indicar el dia i l'hora en el qual s'ha finalitzat l'execució del codi en Python per a la recollida de dades. En cas de no ser així, voldrà dir que no s'ha realitzat amb èxit la connexió amb la Raspberry Pi.





**Figura - 12** Captura de pantalla de la pestanya “Your devices” dintre de la plataforma Edge Impulse.

L'última comprovació que s'ha de fer per assegurar que tot està funcionant amb normalitat és confirmar que les dades realment es poden visualitzar dintre la plataforma Edge Impulse. Per tal de verificar-ho, s'ha d'accedir a la pestanya "Dataset" i allà, tal com mostra la figura [13], hauria d'aparèixer un llistat on es poden visualitzar les etiquetes o classes recollides pel programa en Python.



**Figura - 13** Captura de pantalla de la pestanya “Dataset” dintre la plataforma Edge Impulse

D'aquesta manera, a mesura que es vagin recollint noves classes o etiquetes, es podrà anar veient com es van afegint en el llistat.

### 6.1.2 Descripció del codi de recollida de dades per el model d'AP amb Edge Impulse

A continuació, es fa una explicació detallada del funcionament del codi amb Python per a la recopilació de dades de l'acceleròmetre a la pàgina web Edge Impulse, que es pot trobar en l'annex[VIII]:

- **Importació de les llibreries de Python necessàries:** En el començament del codi es pot apreciar tot un bloc de llibreries importades que permeten fer crida a diferents funcions, classes i variables preprogramades. A continuació es llisten les llibreries:
  - **JSON:** aquesta llibreria permet que les dades en el llenguatge *JSON* (Java script object notation), un llenguatge altament utilitzat en comunicacions web, puguin ser codificades i descodificades en el codi.
  - **Time, hmac i hashlib:** aquestes llibreries de Python es fan servir únicament per a la generació de la firma digital HMAC, que ja he explicat anteriorment en aquest capítol del treball.
  - **Requests:** aquesta llibreria es fa servir per realitzar, juntament amb l'API de la plataforma, sol·licituds HTTP amb la web Edge Impulse.
  - **Re i uuid:** aquestes llibreries serveixen per crear el nom del dispositiu a partir de la direcció MAC de l'adaptador de xarxa de la Raspberry Pi.
  - **Busio, board i adafruit\_adxl34x:** aquestes llibreries fan ús del protocol I2C per tal de crear la comunicació entre l'acceleròmetre ADXL345 i la Raspberry Pi.
- **Definició de constants globals:** En la part superior del codi es declaren tres constants de summa importància, ja que sense elles no es podria realitzar exitosament la comunicació entre la Raspberry Pi i el projecte creat dins la plataforma d'Edge Impulse. Per aquest motiu les constants *HMAC-KEYS*, *API-KEYS* i *INTERVAL\_MS* han de ser declarades al principi del codi. Anteriorment, ja he explicat que són les *HMAC-KEYS* i l'*API-KEYS*, així

que només parlaré en aquest punt de l'INTERVAL\_MS, que és la freqüència de mostreig en mil·lisegons i que compleix el teorema de Nyquist-Shannon (ja explicat en l'apartat [4.4.2])

- **Funcions auxiliars:** en aquest punt, s'especifiquen diverses funcions auxiliars amb un paper específic cada una d'elles:
  - **get\_device\_name():** aquesta funció auxiliar, a partir de la direcció MAC de l'adaptador de xarxa de la Raspberry Pi, genera un identificador únic que permet que Edge Impulse reconegui el dispositiu posteriorment.
  - **create\_adxl345\_instance():** en primer lloc, configura la comunicació I2C amb el sensor. A continuació, crea una instància en el sensor Adafruit ADXL345 per tal de recol·lectar les dades de l'acceleròmetre.
  - **collect\_sensor\_data(accel, interval\_ms, duration):** aquesta, recull les lectures del sensor durant un període de temps determinat per la variable *duration*, una variable que marca els cicles de recol·lecció del programa, conjuntament amb la variable *Interval\_ms*, que marca la freqüència de mostreig de les lectures de l'acceleròmetre. Les dades s'aniran guardant gràcies a un bucle *for* en una llista anomenada *values\_list*. Finalment, aquesta funció auxiliar retornarà la llista *values\_list* amb totes les lectures de l'acceleròmetre Adafruit ADXL345 durant el temps determinat per *duration*.
  - **create\_data\_payload(accel, interval\_ms, device\_name):** aquesta funció auxiliar s'encarrega d'estructurar les dades per ser enviades a Edge Impulse. Aquesta tasca també es coneix en el món de la programació com a *payload*, tal com indica el nom de la funció. En aquest payload aniran les metadades més rellevants per ser enviades a la plataforma. Aquesta estructura, tot i que pugui semblar molt similar a un objecte en JSON, és un diccionari de Python.
  - **sign\_data(data, hmac\_key):** s'encarrega de firmar amb el *hmac\_key* el payload del diccionari de Python *data*. La firma es realitzarà amb l'algoritme

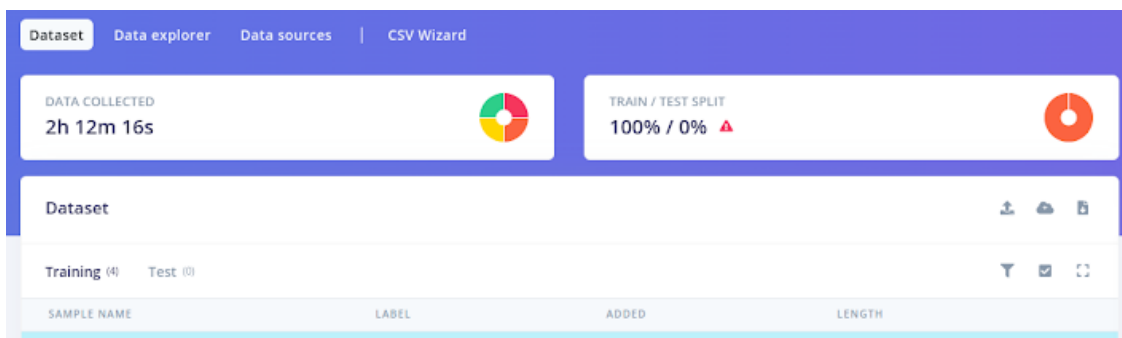
*SHA256* i es converteix en una cadena hexadecimal utilitzant *hexdigest()*. Posteriorment, es guarda en la variable *signature*. Finalment, aquesta funció firma el diccionari de Python *data* que seguidament es convertirà en una cadena de *JSON* amb *json.dumps(data)*. Aquest últim pas serà el retorn de la funció. La única finalitat que té aquesta funció és garantir la seguretat i la integritat de l'enviament de les dades a Edge Impulse.

- **upload\_data(encoded\_data, api\_key):** Aquesta funció es recolza en el mòdul *requests* per a llançar una sol·licitud *POST* als servidors d'Edge Impulse. Tot el que es mana segueix el format *JSON* i incorpora una càrrega de dades degudament signada. Per a poder autenticar la sol·licitud i vincular-la a un compte específic d'Edge Impulse, s'introdueix en les capçaleres la clau *API* que proporciona la mateixa plataforma. No sols això, també s'afegeix un nom d'arxiu *x-file-name* que ajuda a distingir entre els diversos conjunts de dades que s'envien a Edge Impulse. Com a resultat, la funció retorna la resposta que emet el servidor, una cosa molt útil per a poder fer seguiment i comprovar que tot ha anat com la seda.
- **Funció principal:** la funció principal, anomenada *main()*, controla el flux del programa. Aquesta funció és la que duu a terme els passos per rebre les dades del sensor, donant-los format i enviant-los a Edge Impulse.
  - Primer que res, s'ha de verificar que el valor *INTERVAL\_MS* sigui més gran que zero, perquè de no ser així, no tindria sentit una freqüència de mostreig igual a zero o menor, per tant, es llençaria una excepció.
  - Seguidament, es crea el nom del dispositiu amb la funció *get\_device\_name()* i amb la funció *create\_adxl345\_instance()* es crea una instància de l'acceleròmetre ADXL345. Després, *create\_data\_payload()* generarà la *payload* de les dades, que inclou la recollida de les dades de l'acceleròmetre.

- Un cop realitzat l'anterior pas, amb la funció *sign\_data()* es firma la *payload* amb la clau HMAC i així es garanteix l'autenticitat de les dades. Amb aquest procés es crearà una cadena *JSON* que es guarda en la variable *encoded\_data*.
- Per acabar, amb la funció *upload\_data ()* s'envia a Edge Impulse la cadena *encoded\_data*. L'èxit o fracàs de l'operació es veurà en un missatge imprès en el terminal, que dependrà del codi d'estat de la resposta HTTP (200 Ok, 404 Not Found, 500 Internal Server Error).
- **Execució del Programa:** en la línia final del codi, el *main()* només es posa en marxa si el script opera com a arxiu principal, és a dir, no s'incorpora des d'un altre script. Aquest protocol és bastant comú en les aplicacions de Python, i s'utilitza per a habilitar o inhibir l'execució de determinades parts del codi quan els mòduls són importats. Aquesta precaució ens permet mantenir un control més rigorós sobre com i quan s'executen uns certs fragments del codi.

### 6.1.3 Separació de les dades en grups d'entrenament i prova

Tal com mostra a la figura [14], el 100% de les dades estan destinades a training (és a dir, a entrenar el model). El mateix Edge Impulse mostra una alerta al costat d'aquest número perquè el recomanable és destinar part d'aquestes dades al testing (és a dir, a l'avaluació del model). La proporció correcta seria que el 80% de les dades estigues destinada a training i el 20% restant a testing, d'aquesta manera es podria avaluar la precisió del model, de la mateixa manera que s'ha fet amb el model d'AA.



**Figura - 14** Captura de pantalla de la pestanya “Dataset” on es mostra l’alerta en el Train/Test Split dintre la plataforma Edge Impulse

Per separar les dades es farà ús de l'eina split que proporciona Edge Impulse. En la següent figura [15] es pot veure el split realitzat per a cada classe:



**Figura - 15** Captura de pantalla de la pestanya “Split sample” on es mostra el split realitzat Train/Test a la plataforma Edge Impulse

Les dades recol·lectades que es veuen a la plataforma Edge Impulse es quantifiquen per durada en el temps. D'aquesta manera, en la figura [13] es pot veure com cada velocitat té una durada de dades de 33 minuts i 4 segons. El split entre dades per training i dades per testing a la plataforma Edge Impulse es determina en mil·lsegons (ms), per tant, el primer que s'ha de fer és convertir en mil·lsegons el temps de durada de les dades recol·lectades.

Per calcular els mil·lsegons mencionats anteriorment, es fan les següent operacions:

$$1 \text{ minut} = 60,000 \text{ mil} \cdot \text{lsegons}$$

$$1 \text{ segon} = 1,000 \text{ mil} \cdot \text{lsegons}$$

$$33 \text{ minuts} = 33 \cdot 60,000 = 1,980,000 \text{ mil} \cdot \text{lsegons}$$

$$4 \text{ segons} = 4 \cdot 1,000 = 4,000 \text{ mil} \cdot \text{lsegons}$$

$$1,980,000 \text{ ms} + 4,000 \text{ ms} = 1,984,000 \text{ mil} \cdot \text{lsegons}.$$

Per tant, 33 minuts i 4 segons són equivalents a 1,984,000 mil·lisegons. Ara, s'haurà de calcular el 80% i 20% d'aquests mil·lisegons.

Per calcular el 80%:

$$1,984,000 \text{ ms} \cdot 0.80 = 1,587,200 \text{ ms}$$

Per calcular el 20%:

$$1,984,000 \text{ ms} \cdot 0.20 = 396,800 \text{ ms}$$

En la plataforma Edge Impulse, dins de la mateixa pestanya dataset, s'ha d'escollir les dades de l'etiqueta on es vol fer el split. Amb l'opció *Split Sample* es defineix que 1,587,200 ms d'aquestes dades són per training i 396,800 ms són per testing.

Un cop el split està fet, el 20% de les dades de cada velocitat es mourà a testing. Ara la plataforma no donarà cap alerta (com es pot veure a la figura [16]) i el model es podrà testejar adequadament, igual que es va fer amb el model d'AA..



**Figura - 16** Captura de pantalla de la pestanya “Dataset” on ja no es mostra l’alerta en el Train/Test Split dintre la plataforma Edge Impulse

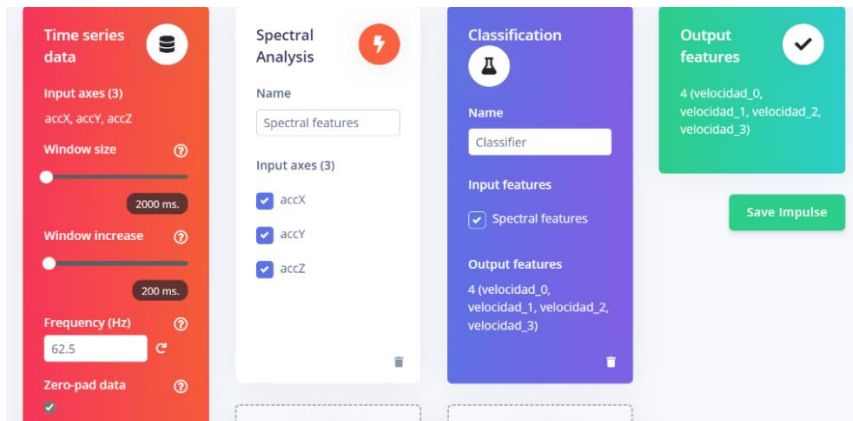
Seguidament, un cop la separació de dades per a training/testing està feta, s'haurà de crear el disseny general de com es vol que sigui el model d'AP.

## 6.2 Disseny general del model d'AP amb Edge Impulse

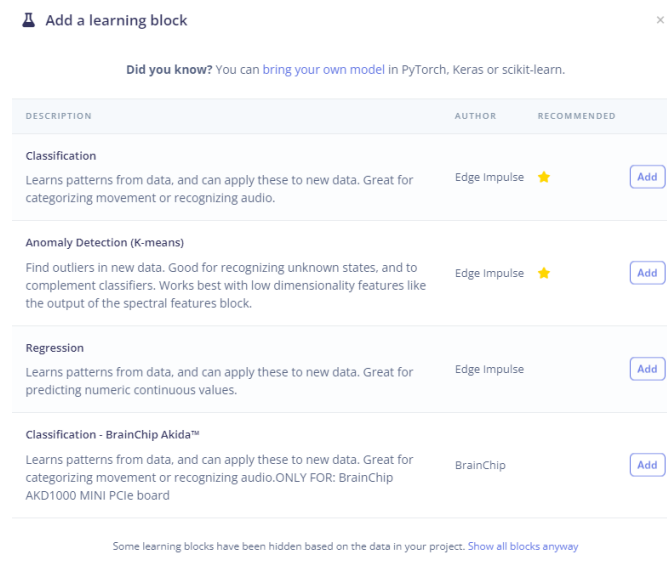
En aquest apartat, es crearà el layout del model d'AP que es vol crear de forma genèrica. El layout el podem trobar en la següent figura [17]. Tal com es pot apreciar, en aquesta pestanya dintre la plataforma, se seleccionen diferents blocs de processament i aprenentatge per tal de configurar el que millor s'adapta al projecte. A continuació es fa una explicació de cada bloc:

- **Time series data:** en aquest bloc es fa referència a la seqüència de punts de les dades recollides pel programa en la pestanya “data acquisition”, en aquest cas les dades recollides per l'acceleròmetre en els eixos X, Y i Z.
  - **Windows size:** està definit a 2000 ms que és la finestra de dades que es pren per a cada mostra.
  - **Window increase:** està definit a 200 ms, això fa referència a quant es desplaça la finestra després de cada mostra.
  - **Frequency (Hz):** La freqüència indica quantes vegades es mesuren o es recullen les dades per segon. En aquest cas serien 62.5 mostres per segon.
  - **Zero-pad data:** és una tècnica utilitzada per fer que totes les mostres tinguin la mateixa longitud de punts de mostreig afegint zeros.
  
- **Bloc de preprocessament:** en aquest bloc, en cas que estigui present en el layout, es fa el preprocessament de les dades, que es poden seleccionar per si només es volen processar algunes d'elles. En aquest cas, s'utilitza aquest bloc per processar totes les dades dels eixos d'acceleració X, Y i Z. El tipus d'anàlisi espectral es definirà més endavant.
  
- **Bloc d'aprenentatge:** en aquest bloc s'escull quin tipus d'aprenentatge es vol realitzar en el projecte. En aquest cas he escollit *Classification*, però hi ha altres opcions tal com mostra la figura [18]. *Classification* és un model d'aprenentatge profund que funciona amb llibreries de TensorFlow. Com es pot veure a la figura [17], com a característiques d'entrada he marcat la casella de característiques espectrals, ja que el model s'entrenarà amb les dades preprocessades que surtin del bloc d'anàlisi espectral.
  
- **Funcions de sortida:** finalment es troba aquest bloc que ens marca les 4 sortides que tindrà el model d'aprenentatge. En aquest cas seran les 4 etiquetes del funcionament normal del ventilador (*velocitat\_0*, *velocitat\_1*, *velocitat\_2* i *velocitat\_3*).





**Figura - 17** Captura de pantalla de la pestanya disseny general del model amb la plataforma Edge Impulse



**Figura - 18** Captura de pantalla del llistat de learning blocks disponibles a Edge Impulse.

Un cop s'ha creat el layout del model d'aprenentatge profund que més s'ajusta al projecte, ja es pot saltar al següent pas i configurar quin tipus d'anàlisi espectral es vol utilitzar tal com s'explica en el següent capítol.

## 6.3 Preprocessament i caracterització espectral de les dades amb Edge Impulse

El preprocessament i la caracterització espectral de les dades recollides per l'acceleròmetre, realitzats en la plataforma Edge Impulse, es presenten com a etapes crítiques en aquest estudi. Aquests processos permeten la transformació de les dades brutes en una representació que sigui més accessible i reveladora de les característiques inherents als senyals recollits. Al llarg d'aquest apartat, descriuré detalladament els paràmetres de configuració emprats per al preprocessament i anàlisi espectral de les dades, així com la fonamentació teòrica que dona suport a aquestes decisions.

### 6.3.1 Filtratge de les Dades amb Edge Impulse

La primera etapa del preprocessament de les dades en Edge Impulse és el filtratge d'aquestes. Aquest pas és molt important per minimitzar l'impacte que les alteracions externes que podrien tenir sobre el senyal. Alguns factors com el soroll extern o les vibracions de cossos aliens al ventilador podrien afectar el senyal, per això és vital reduir aquest impacte al màxim. El filtre que es fa servir en Edge Impulse és un filtre digital IIR (Infinite Impulse Response) i té el benefici que necessita menys potencia de computació comparat amb l'alternativa, FIR (Finite Impulse Response).

Els paràmetres per a la configuració del filtre (tal com mostra la figura [19]) són els següents:

- **Escala d'eixos (Scale axes):** aquest paràmetre es defineix en 1, això vol dir que les dades recollides de l'acceleròmetre no s'escalen (al contrari del que s'ha fet en el preprocessament de les dades pel model d'aprenentatge automàtic en el capítol [5.2]). En el cas d'aquest model, s'ha decidit provar de deixar les dades en la seva amplitud original sense escalar, per comprovar si hi ha alguna diferència rellevant amb la metodologia utilitzada amb l'anterior model (AA).
- **Relació de decimació d'entrada (Input decimation ratio):** aquest paràmetre es fixa en 1, cosa que significa que es reté la totalitat de les dades recollides, sense reduir la quantitat d'informació. Mantenir totes les mostres és important per a conservar la precisió en la representació del senyal. A vegades es fan reduccions de les dades per

alleugerir la càrrega de processament del model, però en aquest cas es prefereix tenir més precisió.

- **Tipus de filtre (Type):** s'opta per un filtre de pas baix (low-pass), amb l'objectiu d'eliminar les altes freqüències que podrien correspondre a soroll o a vibracions alienes al ventilador. En el nostre context, aquestes freqüències altes són d'escàs interès, ja que el rang de freqüències associat a les RPM del ventilador se situa en un rang baix.
- **Freqüència de tall (Cut-off frequency):** es tria una freqüència de tall de 3 Hz. Tenint un filtre low-pass i la freqüència de tall de 3Hz, qualsevol senyal amb una freqüència major a 3Hz s'atenuarà.
- **Ordre del filtre (Order):** s'estableix en 4. L'ordre del filtre està relacionat amb la complexitat amb la qual es fa el càlcul al filtratge. En aquest TFG s'ha decidit que el valor d'ordre de filtre 4 proporciona una transició raonablement abrupta entre la banda de pas i la banda de parada, alhora que manté una complexitat computacional raonable. L'elecció d'un ordre superior podria generar una transició més abrupta, però a costa d'un augment de la complexitat computacional i d'un potencial increment de la distorsió de fase. La selecció de l'ordre 4 representa, per tant, un compromís entre l'atenuació desitjada i les limitacions pràctiques.

### 6.3.2 Anàlisi Espectral de les dades amb Edge Impulse

Un cop finalitzat l'ajust dels paràmetres de filtratge es continuarà amb l'elaboració d'una anàlisi espectral de senyal. Gràcies a aquesta anàlisi s'obtindrà més detall de les característiques importants de les dades recollides, ja que amb ella es pot descompondre un senyal en components.

En el model d'AP amb Edge Impulse, s'ha optat per la Transformada Ràpida de Fourier, coneguda per les seves sigles en anglès FFT.

La FFT converteix una seqüència de dades en el domini del temps en una seqüència de dades en el domini de la freqüència. Cada punt en el domini de la freqüència representa l'amplitud i la fase d'una ona sinusoidal a una freqüència específica. La superposició d'aquestes ones sinusoidals reconstrueix el senyal original en el domini del temps.

La configuració seleccionada per a l'anàlisi espectral dins la plataforma Edge Impulse (tal com mostra la figura [19]) és la següent:

- **Tipus d'anàlisi (Type):** en aquest desplegable s'escull FFT, ja que tal com he explicat és el tipus d'anàlisi espectral que es farà servir en aquest model. Per tant, com he remarcat, amb la FFT es podran identificar els components de freqüència dominants en el senyal, que podrien estar associades a la freqüència de rotació del ventilador o a la freqüència de vibracions generades per les anomalies.
- **Longitud de la FFT (FFT length):** en aquest TFG he decidit que el valor sigui de 16, això indica que el senyal es pot descompondre en 16 components de freqüència. S'ha pres aquesta decisió perquè, tot i que un número més alt hagués permès diferenciar més components, també faria que l'algorisme fos molt més complex. Buscant trobar el balanç entre el detall i el rendiment s'ha establert 16 com la quantitat equilibrada.

The screenshot shows the configuration interface for filtering and spectral analysis in Edge Impulse. It is divided into two main sections: 'Filter' and 'Analysis'.

**Filter Section:**

- Scale axes: 1
- Input decimation ratio: 1
- Type: low
- Cut-off frequency: 3
- Order: 4

**Analysis Section:**

- Type: FFT
- FFT length: 16
- Take log of spectrum?:
- Overlap FFT frames?:
- Improve low frequency resolution?:

A 'Save parameters' button is located at the bottom right of the configuration area.

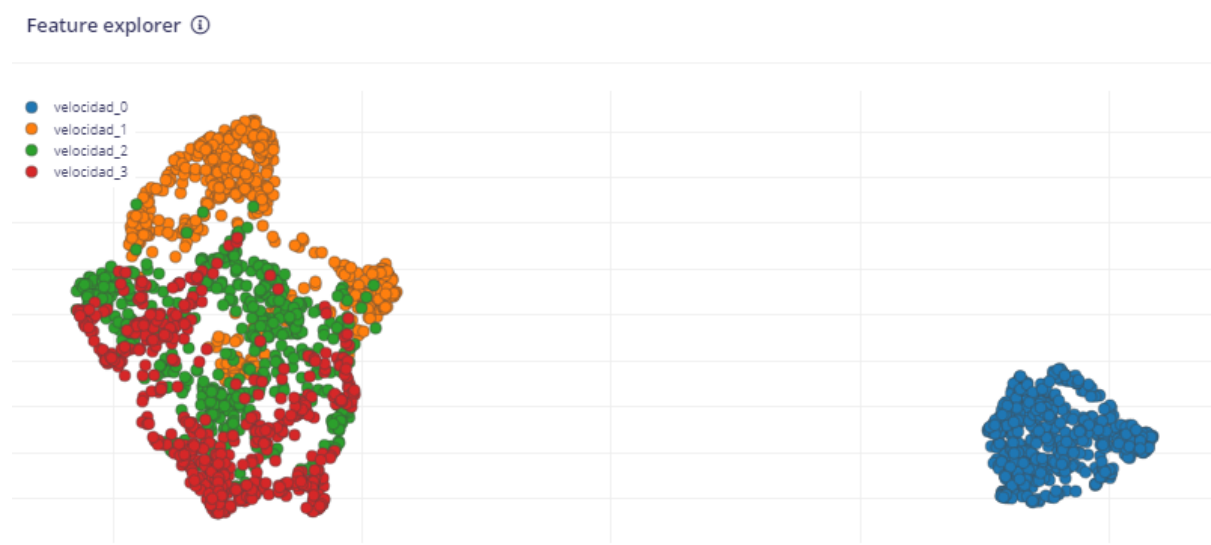
**Figura - 19** Captura de pantalla de la configuració dels paràmetres pel filtratge i l'anàlisi espectral de les dades amb Edge Impulse.

L'aplicació dels processos de filtratge i anàlisi espectral que s'han mencionat en aquest apartat permet extreure informació significativa de les dades recollides per l'acceleròmetre. Aquestes dades processades, ara en un format més accessible i significatiu, em facilitaran l'entrenament del model d'aprenentatge profund, on gràcies a aquest processament de les dades hauria de veure un augment del rendiment en aquest.

### 6.3.3 Gràfic de dispersió de les dades preprocessades

Un cop fet el preprocessament de les dades recollides (és a dir, filtratge i anàlisi espectral) la mateixa plataforma Edge Impulse mostra una representació gràfica de la dispersió d'aquestes, com es pot veure a la figura [20]. A la mencionada figura es pot veure com s'agrupen en el gràfic les diferents classes o etiquetes, per exemple, la classe de la velocitat 0 està clarament separada de la resta. Les Velocitats 1, 2 i 3, al contrari, estan més disperses i els punts se superposen entre ells, sobretot en les velocitats 2 i 3. Per tant, es pot concloure

que el model podrà predir amb precisió la Velocitat 0, tindrà petites dificultats per classificar la velocitat 1 i finalment tindrà més obstacles per predir les velocitats 2 i 3.



**Figura - 20** Gràfic de dispersió de les dades preprocessades per a cada classe amb Edge Impulse.

#### 6.4 Entrenament del model d'AP amb Edge Impulse

Ara que les dades estan preprocessades, es pot procedir a l'entrenament del model d'aprenentatge profund que es farà amb la plataforma Edge Impulse. En la fase de disseny del model [6.2] s'ha decidit escollir un algorisme d'aprenentatge profund anomenat *Classification*, aquest és un model basat en la llibreria open source Tensor Flow. En aquest algorisme, s'hauran de definir alguns paràmetres que s'expliquen a continuació i es mostren en la figura [21]:

- **Configuració de l'entrenament:**
  - **Nombre de cicles d'entrenament (Number of training cycles):** aquest paràmetre defineix les vegades que el model processa totes les dades, en cada cicle (epoch) el model intentarà millorar el processament que fa per obtenir resultats més precisos. Una major quantitat de cicles farà que el model sigui més precís, però existeix el risc de sobreajust. El sobreajust passa quan el model està tan adaptat a les dades d'entrenament que no és capaç de predir

noves dades correctament. Per això és important trobar un equilibri en la quantitat de cicles.

En el cas d'aquest TFG s'ha establert que el nombre de cicles d'entrenament sigui 30, ja que es considera una xifra raonable en el món de la intel·ligència artificial.

- **Taxa d'aprenentatge (learning rate):** aquest paràmetre indica la rapidesa amb la qual el model aprèn, és a dir, la rapidesa del model per canviar els paràmetres que fa servir en cada cicle d'aprenentatge. Un valor alt en aquesta taxa farà que el model aprengui més ràpidament, però de manera menys precisa. Al contrari, un valor més baix voldrà dir que el model és més lent aprenent, però ho farà de manera més precisa i curosa. En aquest cas, s'ha optat per una xifra relativament baixa (0.0005), ja que es busca la precisió en les prediccions del model.
- **Configuració avançada de l'entrenament:**
  - **Mida del conjunt de validació (validation set size):** en aquesta opció es determina la quantitat de dades que el model destinarà a la validació de l'entrenament. En el procés d'entrenament del model d'Edge Impulse, es separa una part de les dades per dedicar-les a la validació. En la validació el model adapta els paràmetres que està fent servir per prevenir el sobreajust i provar el seu rendiment. En aquest TFG s'ha decidit que el valor d'aquest paràmetre sigui el 20%.
  - **Divisió del conjunt d'entrenament/validació en clau de metadades (Split train/validation set on metadata key):** aquesta opció se seleccionarà si no es vol que la separació entre dades per l'entrenament i la validació sigui aleatòria. En el cas d'aquest TFG, no s'ha seleccionat, ja que es vol que ho sigui.
  - **Conjunt de dades autoequilibrat (Auto-balanced dataset):** Si aquesta opció se selecciona, els conjunts de dades per cada classe o etiqueta s'equilibraran si tenen alguna desigualtat. És a dir, si alguna classe està sobrerrepresentada, es

farà un equilibri perquè totes les classes ho estiguin per igual. En aquest TFG s'ha optat per aquesta opció, tot i que les classes estiguin bastant equilibrades, com a mesura preventiva.

- **Perfil de model int8 (profile int8 model):** seleccionant aquesta opció s'obté un model més optimitzat que pot resultar útil, ja que requereix menys memòria i potència de computació. En aquest cas s'ha seleccionat aquesta opció.
- **Arquitectura de les xarxes neuronals (neutral network architecture):** en aquest camp es definiran les diferents capes neuronals pel model d'aprenentatge profund.
  - **Capa d'entrada (input layer):** aquesta capa és la primera en la xarxa neuronal i serà la que rep les dades d'entrada. Edge Impulse configura aquest paràmetre després del preprocessament de les dades, on en aquest cas s'ha extret que hi ha 57 característiques.
  - **Capes denses (dense layer):** aquestes són capes ocultes dins de la xarxa neuronal, cada una té una quantitat de neurones que a la vegada estan també connectades amb la capa anterior. Les capes denses són les que han d'aprendre i detectar els patrons de les dades. En aquest cas hi ha dues capes denses, una de 20 neurones i seguidament una de 10 neurones. Si les dades no presenten patrons massa complexos, aquesta arquitectura hauria de funcionar correctament en el model.
  - **Capa de sortida (output layer):** aquesta capa és la capa final de l'arquitectura de la xarxa neuronal, l'encarregada de predir les classes segons cada dada que rep. En aquest cas les classes serien 4, les diferents velocitats del ventilador (Velocitat 0, 1, 2 i 3) per tant, el valor d'aquesta capa de sortida és de 4.



Neural Network settings

Training settings

Number of training cycles ② 30

Learning rate ② 0.0005

Advanced training settings

Validation set size ② 20 %

Split train/validation set on metadata key ②

Auto-balance dataset ②

Profile int8 model ②

Neural network architecture

Input layer (57 features)

Dense layer (20 neurons)

Dense layer (10 neurons)

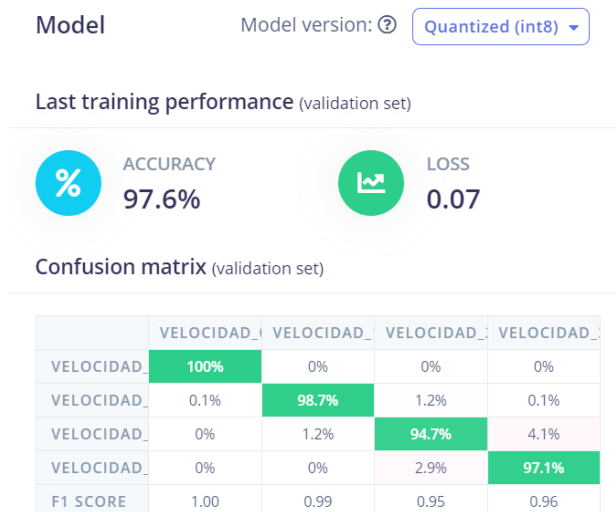
Add an extra layer

Output layer (4 classes)

**Figura - 21** Paràmetres per a l'entrenament del model d'AP amb la plataforma Edge Impulse.

## 6.5 Validació del model d'AP amb Edge Impulse

Tal com s'ha mencionat a l'apartat [6.4], després de l'entrenament el model es realitza un procés de validació per comprovar el seu rendiment. Aquesta validació es mesura amb diferents mètriques que proporcionaran una anàlisi de l'efectivitat i eficiència del model. Igual que passa amb el model d'aprenentatge automàtic, explicat durant el capítol [5], en aquest cas també s'obté una matriu de confusió on es veuen tots els resultats per les diferents classes (tal com es mostra en la figura [22]).



**Figura - 22** Matriu de confusió del model d'AP amb Edge Impulse.

### 6.5.1 Exactitud i Pèrdua del Model

L'Exactitud, igual que en el model d'AA, és la proporció de resultats correctes entre tots els resultats que ha donat el model.

$$Exactitud = \frac{\Sigma TP}{\Sigma TP + \Sigma FP + \Sigma FN}$$

La plataforma Edge Impulse indica que l'exactitud d'aquest model és de 97,6%, una xifra molt satisfactòria, ja que indica que el model és molt precís, sobre tot comparat amb l'exactitud de l'anterior model d'AA que era d'un 60%.

La Pèrdua, per altre banda, indica quina desviació de la realitat tenen les prediccions errònies fetes pel model. El que interessa és que la pèrdua sigui del mínim valor possible perquè això vol dir que el model és molt precís. En aquest cas la pèrdua és del 0,07, que és una molt bona xifra.

### 6.5.2 Anàlisi de la Matriu de Confusió

Tal com es va fer amb el model d'AA en el capítol [5.3.3], la plataforma Edge Impulse mostra una matriu de confusió on es poden veure els percentatges de totes les prediccions del model,

correctes i incorrectes. Passant el ratolí de l'ordinador per sobre d'aquesta matriu es podran veure les quantitats concretes de prediccions en les quals estan basats aquests percentatges.

En aquest cas la velocitat 0 es detecta amb un 100% de precisió, això és una estadística perfecta, cada vegada que el ventilador estava en velocitat 0 el model l'ha detectat a la perfecció. La detecció de la resta de velocitats també és molt bona, ja que s'ha obtingut un 98,7% de precisió en la Velocitat 1, un 94,7% en la velocitat 2 i d'un 97,1% en la velocitat 3. Tot i que hi hagi hagut alguna predicció incorrecta, són molt bones xifres de validació del model.

### **6.5.3 Puntuació F1**

Tal com s'explica en els fonaments teòrics (capítol [3.7.1]), la F1-score té en compte tant la Precisió (el percentatge de prediccions positives correctes entre totes les prediccions positives per cada classe) com el Recall (la proporció de prediccions positives correctes entre tots els positius per cada classe). La F1-score fa una mitjana harmònica de tots dos conceptes.

Els valors de F1 que ha tret aquest model són també molt gratificants, ja que, continuant amb la tònica d'abans, a la velocitat 0 la F1 score és de 1.00, a la velocitat 1 de 0.99, a la velocitat 2 de 0.95 i a la velocitat 3 de 0.96. Això no fa més que reforçar el resultat exitós del model.

### **6.5.4 Gràfica de dispersió posterior a l'entrenament de les dades**

La plataforma Edge Impulse mostra una gràfica de dispersió de les dades ja entrenades. A continuació es pot veure la figura [23] resultant.



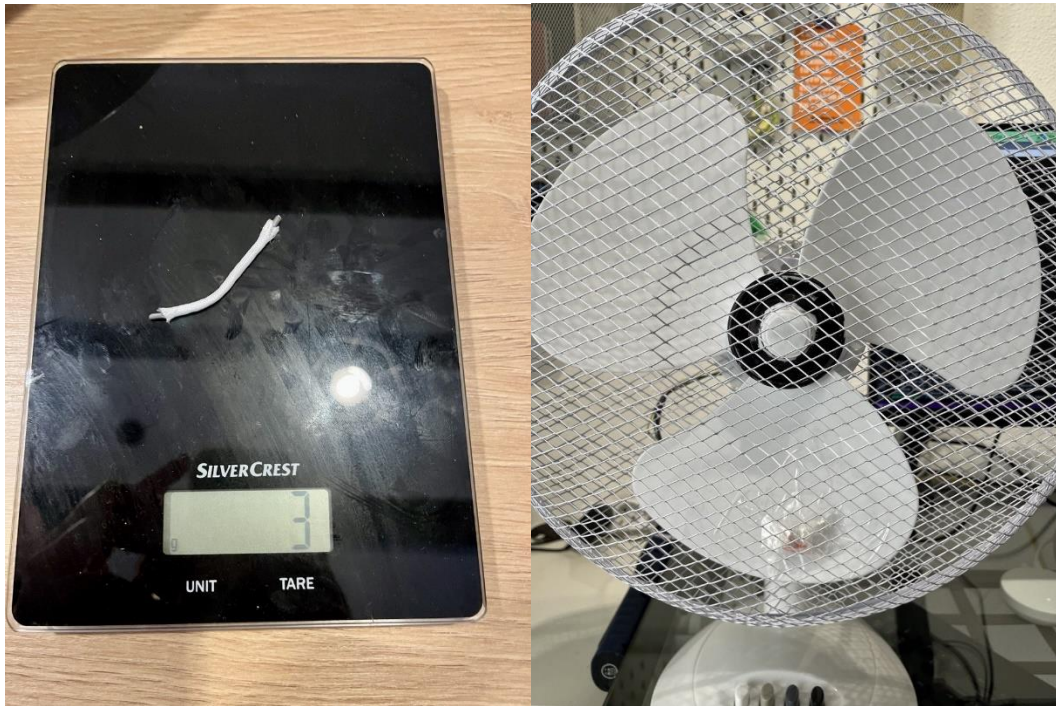
**Figura - 23** Captura de pantalla del gràfic de dispersió de les dades ja entrenades del model d'AP amb Edge Impulse

Com es pot apreciar a la figura [23], en la llegenda les etiquetes apareixen amb el nom "correct", això vol dir que el model les ha predit correctament, en cas contrari s'indicaria amb la paraula "incorrect". En comparació a la figura [20] de les dades abans de l'entrenament, es pot apreciar que hi ha molts menys punts en el gràfic de dispersió, això indica que el model ha creat clusters (agrupacions de les etiquetes) correctament.

## 6.6 Introducció d'una anomalia al model d'AP amb Edge Impulse

Atès que els resultats del model d'aprenentatge profund amb Edge Impulse han sigut tan exitosos, s'ha decidit provar d'introduir una anomalia fora del funcionament normal del ventilador entre les classes, per estudiar com la detecta el model.

La prova es tracta de posar un sobrepès en l'extrem d'una de les aspes del ventilador que afectaria la força centrífuga d'aquest, recollir les dades en les diferents velocitats i observar si el model ho identifica correctament. El pes s'afegeix enganxant un cordill de plom de 3 grams en l'extrem d'una de les aspes del ventilador com mostra la figura [24].

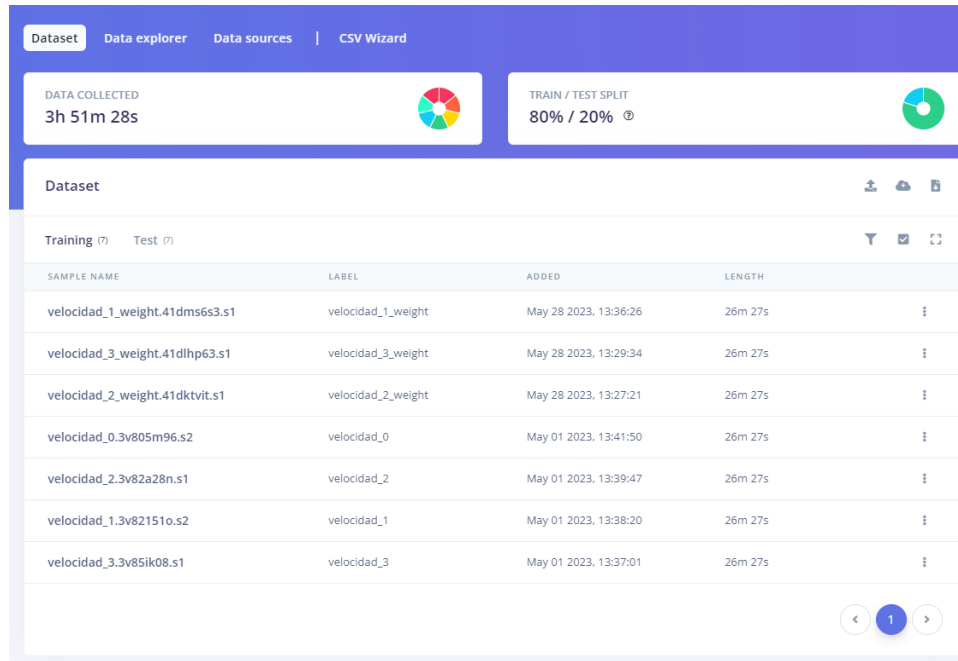


**Figura - 24** Pes i posició del cordill de plom afegit al ventilador per crear una anomalia.

Per trobar el pes exacte de 3 grams es van fer diverses proves amb pesos més alts, però la càrrega era massa gran i l'aspa amb el sobrepès queia cap a baix impeding que el ventilador pogués funcionar. Finalment, vaig decidir que 3 grams era el pes mínim òptim perquè el ventilador funcionés però amb una anomalia.

El sistema per enganxar el pes a l'extrem de l'aspa del ventilador tampoc es va trobar a la primera, ja que, en moltes ocasions, el ventilador llençava disparat el pes a causa de la força centrífuga. Finalment i després de diversos intents, es va aconseguir enganxar el pes de manera segura i estable al ventilador.

Pel procés d'entrenament del model amb l'anomalia, es repetiran els passos mencionats en apartats anteriors, primer es recolliran les dades amb la mateixa metodologia mencionada a l'apartat [6.1] i es preprocessaran segons l'explicat en l'apartat [6.3]. Així, dins del mateix model, hi haurà 7 classes diferents, les 4 velocitats mencionades anteriorment (Velocitat 0, 1, 2 i 3) i altres tres velocitats amb el sobrepès (Velocitat\_1\_weight, Velocitat\_2\_weight i Velocitat\_3\_weight). Tot això es mostra en la següent figura [25].



The screenshot shows the 'Dataset' page in the Edge Impulse interface. At the top, there are navigation tabs: 'Dataset', 'Data explorer', 'Data sources', and 'CSV Wizard'. Below these, there are two summary cards: 'DATA COLLECTED' showing '3h 51m 28s' and 'TRAIN / TEST SPLIT' showing '80% / 20%'. The main content is a table with columns for 'SAMPLE NAME', 'LABEL', 'ADDED', and 'LENGTH'. The table lists seven samples with labels like 'velocidad\_1\_weight', 'velocidad\_3\_weight', etc. At the bottom right, there are navigation arrows and a page number '1'.

SAMPLE NAME	LABEL	ADDED	LENGTH
velocidad_1_weight.41dms6s3.s1	velocidad_1_weight	May 28 2023, 13:36:26	26m 27s
velocidad_3_weight.41dlhp63.s1	velocidad_3_weight	May 28 2023, 13:29:34	26m 27s
velocidad_2_weight.41dktvit.s1	velocidad_2_weight	May 28 2023, 13:27:21	26m 27s
velocidad_0.3v805m96.s2	velocidad_0	May 01 2023, 13:41:50	26m 27s
velocidad_2.3v82a28n.s1	velocidad_2	May 01 2023, 13:39:47	26m 27s
velocidad_1.3v82151o.s2	velocidad_1	May 01 2023, 13:38:20	26m 27s
velocidad_3.3v85ik08.s1	velocidad_3	May 01 2023, 13:37:01	26m 27s

**Figura - 25** Captura de pantalla de la pestanya Dataset amb els valors de l'anomalia recollits a Edge Impulse.

L'entrenament del model es fa de la mateixa manera que es va fer en l'apartat [6.4], l'única cosa que canviarà serà l'output layer (capa de sortida) de l'arquitectura de la xarxa neuronal, ja que passarà de tenir 4 outputs o classes a identificar, a tenir 7 classes. Això es pot veure en la figura [26].

Neural Network settings

Training settings

Number of training cycles 30

Learning rate 0.0005

Advanced training settings

Validation set size 20 %

Split train/validation set on metadata key

Auto-balance dataset

Profile int8 model

Neural network architecture

Input layer (57 features)

Dense layer (20 neurons)

Dense layer (10 neurons)

Add an extra layer

Output layer (7 classes)

**Figura - 26** Captura de pantalla dels paràmetres d'entrenament del model amb l'anomalia a Edge Impulse.

### 6.6.1 Validació del model d'AP amb Edge Impulse introduint l'anomalia

Com s'ha explicat en l'apartat [6.5], en la fase de validació del model, Edge Impulse mostra una matriu de confusió amb tots els resultats, tal com es mostra a la figura [27]. Una vegada més, l'exactitud del model està en números molt bons, un 98,7% per ser exactes. La Pèrdua és molt baixa, seguint la tònica de la prova amb les velocitats del funcionament normal. A més la F1-score de les classes Velocitat\_1\_weight, Velocitat\_2\_weight i Velocitat\_3\_weight és de 1.00, el que indica que el model identifica aquestes classes amb molta precisió.

Last training performance (validation set)



Confusion matrix (validation set)

	VELOCIDAD_0	VELOCIDAD_1	VELOCIDAD_1_WEIGI	VELOCIDAD_2	VELOCIDAD_2_WEIGI	VELOCIDAD_3	VELOCIDAD_3_WEIGI
VELOCIDAD_0	99.9%	0.1%	0%	0%	0%	0%	0%
VELOCIDAD_1	0.1%	98.4%	0%	1.5%	0%	0%	0%
VELOCIDAD_1_WEIGHT	0%	0%	100%	0%	0%	0%	0%
VELOCIDAD_2	0%	1.1%	0%	95.3%	0%	3.6%	0%
VELOCIDAD_2_WEIGHT	0%	0%	0.1%	0%	99.9%	0%	0%
VELOCIDAD_3	0%	0%	0%	3.1%	0%	96.9%	0%
VELOCIDAD_3_WEIGHT	0%	0%	0%	0%	0%	0%	100%
F1 SCORE	1.00	0.99	1.00	0.95	1.00	0.97	1.00

**Figura - 27** Captura de pantalla de la matriu de confusió incloent les classes de l'anomalia en Edge Impulse.

### 6.7 Testing del model d'aprenentatge profund amb Edge Impulse.

Tal com s'ha comentat a l'apartat [6.1.3], s'ha reservat un 20% de les dades recollides per la fase de testing, és a dir, per ferles servir en el model ja entrenat i avaluar el rendiment d'aquest. Com a resultat del testing, es torna a mostrar una matriu de confusió de les prediccions correctes i incorrectes de les dades fetes servir pel testing. Aquesta matriu es pot veure en la figura [28].



	VELOCIDAD_0	VELOCIDAD_1	VELOCIDAD_1_WEI	VELOCIDAD_2	VELOCIDAD_2_WEI	VELOCIDAD_3	VELOCIDAD_3_WEI	UNCERTAIN
VELOCIDAD_0	100%	0%	0%	0%	0%	0%	0%	0%
VELOCIDAD_1	0%	88.3%	0%	7.7%	0%	0.1%	0%	3.8%
VELOCIDAD_1_WEIGHT	0%	0%	100%	0%	0%	0%	0%	0%
VELOCIDAD_2	0%	0.5%	0%	92.2%	0%	5.1%	0%	2.3%
VELOCIDAD_2_WEIGHT	0%	0%	0.0%	0%	99.8%	0%	0.0%	0.1%
VELOCIDAD_3	0%	0%	0%	2.5%	0%	95.9%	0%	1.7%
VELOCIDAD_3_WEIGHT	0%	0%	0%	0%	0.0%	0%	99.9%	0.0%
F1 SCORE	1.00	0.94	1.00	0.91	1.00	0.95	1.00	

**Figura - 28** Captura de pantalla de la matriu de confusió (testing) incloent les classes de l'anomalia en Edge Impulse.

Aquesta vegada l'exactitud, tot i que continua sent una xifra alta i molt bona, ha baixat una mica en comparació amb l'exactitud de la validació. També es pot veure que a l'hora de detectar les velocitats 1,2 i 3 del funcionament normal hi ha hagut més errors. No obstant



això, les velocitats amb el sobrepès (Velocitat\_1\_weight, Velocitat\_2\_weight i Velocitat\_3\_weight) continuen amb una exactitud de gaire bé el 100% i una F1 score de 1.00.

## 7. Resultats i discussió

Examinant en detall tots els resultats, s'observa que en el model d'aprenentatge automàtic s'ha obtingut una exactitud d'un 60%, igual que la mitjana macro i ponderada. En relació amb els resultats de cada classe, el model d'aprenentatge automàtic sí que mostra una alta precisió i bones mètriques per detectar la Velocitat 0. En canvi, baixa molt el rendiment quan es tracta de detectar la resta de velocitats del funcionament normal del ventilador, rondant del 44% al 54% la precisió del model en detectar aquestes classes.

El model d'aprenentatge profund amb Edge Impulse, al contrari, mostra molts bons resultats en el seu rendiment. Aquest arriba a una exactitud del 98,7% en la validació de les 7 classes (velocitats normals de funcionament i velocitat amb sobrepès). Per altra banda, s'assoleix un 96,58% d'exactitud en el testing de les mateixes 7 classes. A més, aquest model també ha tingut molt bons resultats en la precisió per detectar cada classe. La més baixa d'un 95,3% en la validació i d'un 88,3% en el testing, i la més alta d'un 100% en la validació i testing.

Per tant, tal com es conclou a l'apartat de conclusions [8], el model d'Edge Impulse basat en TensorFlow és molt més precís que el model d'aprenentatge automàtic de boscos aleatoris. L'únic inconvenient, que s'ha detectat en Edge Impulse és que no és un sistema gratuït, tot i que sí que ofereix una opció gratuïta, aquesta és limitada a una certa quantitat de dades. En el cas d'aquest TFG, opció gratuïta ha sigut suficient per al desenvolupament del model.

## 8. Conclusions i treballs futurs

### 8.1 Conclusions

L'objectiu d'aquest TFG consistia en estudiar la possibilitat d'implementar un sistema de manteniment predictiu basat en intel·ligència artificial. Per això, s'han desenvolupat dos models d'aprenentatge automàtic i profund que puguin predir les diferents velocitats i una anomalia del funcionament d'un ventilador.

Primerament, s'ha desenvolupat el model d'aprenentatge automàtic basat en boscos aleatoris pensant que seria la solució definitiva per detectar tant velocitats com anomalies del ventilador. No obstant això, em vaig trobar amb diversos problemes durant el camí. Primer de tot, el desenvolupament de codi Python pel model ha sigut molt complicat, ja que partia d'una base elemental de coneixement en programació, però aquesta tasca exigia més. El codi va donar diversos errors pels quals vaig haver de trobar solució un per un.

Va haver-hi obstacles amb la connexió entre la Raspberry Pi 4 i l'acceleròmetre Adafruit ADXL345 amb la llibreria *busio*. Després d'haver executat el codi i en mig del procés de millorar-lo, va sorgir un error on el dispositiu I2C (acceleròmetre) no es detectava, tot i que sí que estava físicament connectat. Aquest error va persistir durant tant de temps que es va haver de comprar nous sensors i refer les connexions soldades del cablejat pensant que el problema era físic, no obstant això, l'error era de la llibreria *busio*.

Un altre error que vaig detectar va ser en el moment d'instal·lar les llibreries en la Raspberry Pi 4, no ho vaig fer en un entorn virtual. Això fa que les llibreries entre si puguin tenir conflictes i atès que en aquest TFG es fa servir una quantitat important de llibreries, podria ser que l'error anterior de la llibreria *busio* pogués ser degut a la falta d'un entorn virtual.

Al visualitzar la recollida de dades en gràfics en temps real, també van sortir complicacions. A vegades els gràfics no mostraven correctament la informació, comportant-se de forma no desitjada i aleatòria.

En la fase de deployment es va voler afegir una secció dins del gràfic on es mostrés el nom de la velocitat detectada en temps real. Com que no trobava la solució per aconseguir això correctament, vaig buscar una solució alternativa més senzilla. Aquesta consistia a imprimir el nom de l'etiqueta o velocitat en el terminal i eliminar el gràfic en temps real.

En executar el model i avaluar el rendiment d'aquest, els resultats no van ser tan precisos com s'esperava, rondant una exactitud del 60%.

Es pot concloure que pel desenvolupament d'un model d'aprenentatge automàtic que s'implementi en un sistema de manteniment predictiu s'han de tenir alts coneixements de programació. En el cas que es volgués implementar en el món real, això implicaria un major cost en personal qualificat.

A partir dels resultats de precisió del model, es va determinar que el model d'aprenentatge automàtic no era la solució més satisfactòria per l'objectiu d'aquest TFG. Vaig decidir provar amb un nou model, aquesta vegada basat en aprenentatge profund i desenvolupat en un entorn més “user-friendly” com és la plataforma Edge Impulse.

En desenvolupar el model d'aprenentatge profund en Edge Impulse, el procés va resultar molt més senzill. L'únic pas més difícil va ser el moment de realitzar la connexió i l'enviament de dades entre la Raspberry Pi 4 connectada a l'acceleròmetre Adafruit ADXL345 i la plataforma Edge Impulse. Per començar, es va haver de desenvolupar un codi que envies les dades a la plataforma, però aquesta només admet les dades en format JSON. El codi es va haver de connectar amb la plataforma a través de les claus API i HMAC del usuari i enviar les dades en l'estructura i format JSON perquè Edge Impulse les acceptes.

Els resultats de les prediccions amb el model d'aprenentatge profund d'Edge Impulse, van ser molt satisfactoris, per tant, es va decidir que aquest model era l'adequat per continuar amb la investigació d'aquest TFG. Tenint en compte que havia detectat amb tanta precisió les velocitats del funcionament normal del ventilador, vaig decidir incorporar una anomalia en les dades.

L'anomalia consistia a afegir un pes extra en un extrem d'una de les aspes del ventilador per veure com influïa en les vibracions. Es van fer proves amb diferents pesos fins a trobar al pes mínim òptim. Un problema que va sorgir en aquest procés, va ser en enganxar el pes a l'aspa i que aquest no surtis disparat a causa de la força centrífuga del ventilador. Després d'unes quantes proves, es va aconseguir que el pes es mantingués a l'aspa.

Després de recol·lectar les dades de les diferents velocitats en el ventilador amb el sobrepès i d'entrenar el model d'aprenentatge profund en Edge Impulse, els resultats van tornar a ser molt bons. El model encara prediu amb precisió les diferents velocitats, sigui amb pes o sense

ell. Sí que és veritat que a l'hora de fer el testing d'aquest model va sortir alguna imprecisió, però l'exactitud del model es manté molt per sobre que la del model d'aprenentatge automàtic.

La conclusió és que, per estudiar la implementació d'un sistema de manteniment predictiu basat en intel·ligència artificial per a un ventilador, el model que dona millors resultats ha sigut el d'aprenentatge profund basat en la llibreria TensorFlow i desenvolupat a la plataforma Edge Impulse.

Com a conclusió general d'aquest TFG, també es pot dir que la intel·ligència artificial ha fet la seva feina. Fent servir aquesta eina s'ha aconseguit l'objectiu d'identificar velocitats i una anomalia del funcionament del ventilador a partir de les vibracions d'aquest. La intel·ligència artificial, no només ha assolit la seva tasca, sinó que ho ha fet amb una precisió per sobre del que és capaç l'ésser humà.

També s'ha de tenir en compte que, sent un estudiant d'enginyeria mecànica sense cap mena d'especialitat en el camp de la intel·ligència artificial, he aconseguit dur a terme aquest estudi. Això m'ofereix com a enginyer un perfil professional més complet. A més, aquest TFG demostra que la intel·ligència artificial és una eina que facilita extremadament l'anàlisi de les dades i té un potencial increïble en moltíssims camps, com pot ser en aquest cas el manteniment predictiu.

No obstant això, hi ha molt desconeixement en l'entorn de la intel·ligència artificial en el món empresarial. Moltes companyies opten per solucions més costoses i a vegades menys eficients, per desconeixement o per por a una eina que està evolucionant de forma exponencial. Malgrat això, amb aquest TFG s'ha demostrat que solucions basades en intel·ligència artificial no són tan inaccessibles. D'aquesta manera, es pot concloure que la intel·ligència artificial és una solució aplicable a quasi tots els camps i fer-la servir o no, més que una qüestió de gran complexitat, és una decisió.

## 8.2 Limitacions i treballs futurs

La primera i més gran limitació amb la qual m'he trobat és la falta de coneixements avançats en l'entorn de la programació. Sobretot a l'hora de desenvolupar el model d'aprenentatge automàtic, tot hauria sigut més fàcil amb aquesta formació. No obstant això, he desenvolupat solucions a tots els problemes que s'han anat presentant.

Un altre limitant per futures recerques, podria ser la capacitat de computació de la Raspberry Pi respecte als models d'intel·ligència artificial. Aquests models funcionen amb tensors, per tant, un ordinador amb processador especialitzat per tensors, també conegut com a TPU (Tensor Processing Unit), facilita el processament dels models. Una bona alternativa a la Raspberry Pi seria, per exemple, la placa de desenvolupament Google Coral Dev Board o NVIDIA Jetson Nano, totes dues amb una TPU incorporada. Una altra solució, podria ser, connectar un accelerador USB de la marca Coral a la Raspberry Pi, d'aquesta manera aportariem una TPU a la Raspberry.

Un altre impediment, de cara a recerques amb més quantitat de dades, seria la limitació de la versió gratuïta d'Edge Impulse. La part gratuïta de la plataforma permet un màxim de 3 persones en el mateix projecte i té un límit de 4 GB o 4 hores de dades per classificar. Per tant, en projectes o maquinària amb moltes més classes i dades és un delimitant. En aquest cas, seria necessari accedir a la subscripció de pagament, amb el que deixaria de ser un sistema de baix cost.

L'alternativa que es proposa a aquest obstacle és fer servir l'Edge Impulse de manera gratuïta a petita escala per veure com s'han d'ajustar certs paràmetres de manera senzilla. Segons el tipus de classificació escollit, l'ideal seria fer un programa basat en aquest tipus de classificació, programat en Python i fent servir TensorFlow i Keras. Aquesta solució s'hauria de dur a terme amb plaques de desenvolupament que incorporin una TPU, tal com he explicat anteriorment.

Per a continuar investigant més profundament en les bases d'aquest TFG, el primer que recomanaria per a treballs futurs, seria fer un desplegament del model en diferents ventiladors TROTEC TVE 14 i veure si el model detecta correctament les diferents velocitats en tots els ventiladors. Això demostra que tots els ventiladors tenen vibracions similars de fàbrica, i seria un pas més per implementar aquest sistema com a solució de manteniment predictiu aplicable

en el món industrial. D'aquesta manera, es podria confirmar que el model té la capacitat de detectar un patró estàndard de la maquinària.

Una altra tasca que es podria realitzar per millorar aquest TFG, seria col·locar diversos sensors a la vegada i creuar les dades d'aquests. Això potser amb una màquina senzilla com és el ventilador no aporta gaire informació nova, però podria ser clau en màquines més complexes. També es podrien incloure sensors de temperatura per ampliar el marc de les dades i tenir una visió més ampla del funcionament de la maquinària.

Si es volgués implementar aquest sistema de manteniment predictiu en el món real, un cop validat el model, seria necessari recopilar moltes més anomalies per tenir una visió més completa de les avaries o errors que pugui tenir la màquina, i així, poder predir amb més exactitud noves incidències i el seu origen (com més dades rellevants, millor).

El que està clar és que la intel·ligència artificial ha vingut per quedar-se, i aviat començarà a ser una eina completament quotidiana en el nostre dia a dia. Qualsevol recerca en aquesta direcció ens acosta a una eficiència màxima.

## 9. Referències


- [1] “Tutorials,” TensorFlow, Sep. 2021, [Online]. Disponible a: <https://www.tensorflow.org/tutorials>
- [2] Emmet, “Raspberry Pi Accelerometer using the ADXL345,” Pi My Life Up, Oct. 2022, [Online]. Disponible a: <https://pimylifeup.com/raspberry-pi-accelerometer-adxl345/>
- [3] “¿Cómo funciona y qué hace el acelerómetro?” Distribuidor De Componentes Electrónicos. Tienda En Línea: Transfer Multisort Elektronik, Oct. 2020, [Online]. Disponible a: <https://www.tme.eu/es/news/library-articles/page/22568/Como-funciona-y-que-hace-el-acelerometro/>
- [4] “User guide: contents,” [Online]. Disponible a: Scikit-learn. [https://scikit-learn.org/stable/user\\_guide.html](https://scikit-learn.org/stable/user_guide.html)
- [5] Fractal, “Tipos de mantenimiento: reactivo y proactivo,” Fractal, Oct. 2022, [Online]. Disponible a: <https://www.fractal.com/es/blog/tipos-de-mantenimiento>
- [6] “Continuous motion recognition - Edge Impulse Documentation.”, [Online]. Disponible a: <https://docs.edgeimpulse.com/docs/tutorials/continuous-motion-recognition>
- [7] “P. Eugen. The GPIO header layout. Source: Raspberry Pi Foundation.” ResearchGate. [Online]. Disponible a: [https://www.researchgate.net/figure/The-GPIO-header-layout-Source-Raspberry-Pi-Foundation\\_fig1\\_342232287](https://www.researchgate.net/figure/The-GPIO-header-layout-Source-Raspberry-Pi-Foundation_fig1_342232287)
- [8] “ADXL345 Digital Accelerometer,” Adafruit Learning System, Mar. 26, 2013, [Online]. Disponible a: <https://learn.adafruit.com/adxl345-digital-accelerometer/assembly-and-wiring>
- [9] “Vibration Analysis for Machinery Health Diagnosis”, EMERSON, [Online]. Disponible a: <https://www.emerson.com/documents/automation/brochure-vibration-analysis-for-machinery-health-diagnosis-ams-en-6652272.pdf>
- [10] E. Burns, “In-depth guide to machine learning in the enterprise,” Enterprise AI, Apr. 2021, [Online]. Disponible a: <https://www.techtarget.com/searchenterpriseai/In-depth-guide-to-machine-learning-in-the-enterprise>



- [11] “Build Deep Learning Models with TensorFlow | Codecademy,” Codecademy. [Online]. Disponible a: <https://www.codecademy.com/learn/paths/build-deep-learning-models-with-tensorflow>
- [12] B. Earl, "Vibration Analysis for Machinery Health Diagnosis", EMERSON, Des. 2022. [Online]. Disponible a: <https://www.emerson.com/documents/automation/brochure-vibration-analysis-for-machinery-health-diagnosis-ams-en-6652272.pdf>
- [13] “Raspberry Pi Documentation.” [Online]. Disponible a: <https://www.raspberrypi.com/documentation/>
- [14] “Build a Machine Learning Model with Python | Codecademy,” Codecademy, [Online]. Disponible a: <https://www.codecademy.com/learn/paths/machine-learning>
- [15] “Getting Started - Edge Impulse Documentation.”, [Online]. Disponible a: <https://docs.edgeimpulse.com/docs/>
- [16] DeepLearning.AI, “Resources - DeepLearning.AI,” DeepLearning.AI, Oct. 19, 2022. [Online]. Disponible a: <https://www.deeplearning.ai/resources/#course-slides>
- [17] X. Li, J.-L. Li, Y. Qu, and D. He, “Semi-supervised gear fault diagnosis using raw vibration signal based on deep learning,” Chinese Journal of Aeronautics, vol. 33, no. 2, pp. 418–426, Feb. 2020, doi: 10.1016/j.cja.2019.04.018. [Online]. Disponible a: <https://www.sciencedirect.com/science/article/pii/S1000936119302018>
- [18] Snmahajan, “vibration-analysis-using-adxl345-python/adxl345\_final.py at master · snmahajan30/vibration-analysis-using-adxl345-python,” GitHub [Online]. Disponible a: [https://github.com/snmahajan30/vibration-analysis-using-adxl345-python/blob/master/adxl345\\_final.py](https://github.com/snmahajan30/vibration-analysis-using-adxl345-python/blob/master/adxl345_final.py)
- [19] J. Hrisko, “Raspberry Pi Vibration Analysis Experiment With a Free-Free Bar — Maker Portal,” Maker Portal, Mar. 2019, [Online]. Disponible a: <https://makersportal.com/blog/2019/2/26/raspberry-pi-vibration-analysis-experiment-with-free-free-bar>
- [20] “Google Colaboratory.” [Online]. Disponible a: [https://colab.research.google.com/github/arunptp/Predictive-Analytics/blob/master/BearingAnalytics\\_.ipynb](https://colab.research.google.com/github/arunptp/Predictive-Analytics/blob/master/BearingAnalytics_.ipynb)

- [21] “Social Network for Programmers and Developers.” [Online]. Disponible a: <https://morioh.com/p/494532ac3395>
- [22] “Raspberry Pi 4 - Edge Impulse Documentation.” [Online]. Disponible a: <https://docs.edgeimpulse.com/docs/development-platforms/officially-supported-cpu-gpu-targets/raspberry-pi-4>
- [23] “Spectral features - Edge Impulse Documentation.” [Online]. Disponible a: <https://docs.edgeimpulse.com/docs/edge-impulse-studio/processing-blocks/spectral-features>
- [24] R. KeepCoding, “¿Qué son las redes neuronales recurrentes?,” KeepCoding Bootcamps, Feb. 2023. [Online]. Disponible a: <https://keepcoding.io/blog/que-son-las-redes-neuronales-recurrentes/>
- [25] “¿Qué son las redes neuronales convolucionales? | IBM.” [Online]. Disponible a: <https://www.ibm.com/es-es/topics/convolutional-neural-networks>
- [26] Data Science Team, “Regularización,” DATA SCIENCE, Maig 2020, [Online]. Disponible a: <https://datascience.eu/es/aprendizaje-automatico/regularizacion-en-el-aprendizaje-automatico/>
- [27] “Aprendiendo Inteligencia Artificial: ¿Qué es la regularización? - Comunidad Huawei Enterprise,” Comunidad Huawei Enterprise, jun. 2023 [Online]. Disponible a: <https://forum.huawei.com/enterprise/es/aprendiendo-inteligencia-artificial-%C2%BFqu%C3%A9-es-la-regularizaci%C3%B3n/thread/745541-100757#:~:text=La%20regularizaci%C3%B3n%20es%20una%20tecnolog%C3%ADa,un%20gran%20n%C3%BAmero%20de%20par%C3%A1metros>
- [28] K. Otness, X. Zhang, S. Chandrakaladharan, C. Raach, “Overfitting and regularization - Aprendizaje Profundo.”, Sep. 2020, [Online]. Disponible a: <https://atcold.github.io/pytorch-Deep-Learning/es/week14/14-3/>
- [29] M. Martínez, “Transformada de Fourier: qué es y cómo se calcula,” Nobbot, Maig 2021. [Online]. Disponible a: <https://www.nobbot.com/que-es-la-transformada-de-fourier-y-para-que-sirve/>
- [30] “¿Que es Raspberry Pi? - Raspberry Pi,” Raspberry Pi, Apr. 2022, [Online]. Disponible a: <https://raspberrypi.cl/que-es-raspberry/>

- [31] J. Guerra, “Comunicación I2C con Arduino lo mejor de 2 mundos,” Programar fácil con Arduino, Jan. 2022. [Online]. Disponible a: <https://programarfácil.com/blog/arduino-blog/comunicacion-i2c-con-arduino/>
- [32] L. Llamas, “Explicación del teorema de muestreo de Nyquist sin ecuaciones,” Luis Llamas, Des. 2020, [Online]. Disponible a: <https://www.luisllamas.es/explicacion-del-teorema-de-muestreo-de-nyquist-sin-ecuaciones/>
- [33] S. Morell, “Aliasing y el teorema de muestreo de Nyquist-Shannon,” Quantum Society, Jun. 2020, [Online]. Disponible a: <https://quantum-society.com/2020/06/21/aliasing-y-el-teorema-de-muestreo-de-nyquist-shannon/>
- [34] A. González, “Conceptos básicos de Machine Learning – Cleverdata.” [Online]. Disponible a: <https://cleverdata.io/conceptos-basicos-machine-learning/>
- [35] “Exactitud y precisión”, AcademiaLab, [Online]. Disponible a: <https://academia-lab.com/enciclopedia/exactitud-y-precision/>
- [36] “¿Qué es el etiquetado de datos? | IBM.” <https://www.ibm.com/es-es/topics/data-labeling>
- [37] “Escalado de datos | Interactive Chaos.” [Online]. Disponible a: <https://interactivechaos.com/es/manual/tutorial-de-machine-learning/escalado-de-datos>
- [38] J.C. Tovar, “Aprendizaje Automático-Label Encoding en Python - Comunidad Huawei Enterprise,” Comunidad Huawei Enterprise. Sep. 2022, [Online]. Disponible a: <https://forum.huawei.com/enterprise/es/aprendizaje-autom%C3%A1tico-label-encoding-en-python/thread/997385-100757#:~:text=El%20label%20encoding%20se%20refiere,se%20deben%20operar%20esas%20etiquetas>
- [39] Dpto Matematica Aplicada, “Ratios e indicadores que califican una clasificación”, Universidad Politécnica de Madrid, [Online]. Disponible a: <https://dcain.etsin.upm.es/~carlos/bookAA/02.3MetodosdeClasificacion-Ratioseindicadores.html>
- [40] “¿Qué es un bosque aleatorio? | IBM.” [Online]. Disponible a: <https://www.ibm.com/mx-es/topics/random-forest>

- [41] “Random Forest: Bosque aleatorio. Definición y funcionamiento,” Formation Data Science | DataScientest.com, Gen. 2022. [Online]. Disponible a: <https://datascientest.com/es/random-forest-bosque-aleatorio-definicion-y-funcionamiento>
- [42] “Machine Learning: Seleccíon Métricas de clasificaci3n - sitiobigdata.com,” sitiobigdata.com, Oct. 27, 2019. [Online]. Disponible a: <https://sitiobigdata.com/2019/01/19/machine-learning-metrica-clasificacion-parte-3/#>
- [43] “3.3. Metrics and scoring: quantifying the quality of predictions,” Scikit-learn. [Online]. Disponible a: [https://scikit-learn.org/stable/modules/model\\_evaluation.html](https://scikit-learn.org/stable/modules/model_evaluation.html)
- [44] GeeksforGeeks, “Machine Learning Model Evaluation,” GeeksforGeeks, Jan. 2023, [Online]. Disponible a: <https://www.geeksforgeeks.org/machine-learning-model-evaluation/>
- [45] L. Gonzalez, “Métricas de Evaluaci3n Clasificaci3n con Scikit Learn,”  Aprende IA, Oct. 2020, [Online]. Disponible a: <https://aprendeia.com/metricas-de-evaluacion-clasificacion-con-scikit-learn-machine-learning/>
- [46] W. Barrios, “Calculando la precisi3n en un modelo de Clasificaci3n Multiclase,” Medium, Jan. 2022. [Online]. Disponible a: <https://wbarriosb.medium.com/calculando-la-precisi%C3%B3n-en-un-modelo-de-clasificaci%C3%B3n-multiclase-224d96f52043>
- [47] “Claves HMAC,” Google Cloud, Nov. 2022, [Online]. Disponible a: <https://cloud.google.com/storage/docs/authentication/hmackeys?hl=es-419>
- [48] “Resultados de los modelos de aprendizaje autom3tico - Finance | Dynamics 365,” Microsoft Learn, Mar. 2023. [Online]. Disponible a: <https://learn.microsoft.com/es-es/dynamics365/finance/finance-insights/confusion-matrix>
- [49] Evaluaci3n de los resultados de los experimentos de aprendizaje autom3tico automatizado - Azure Machine Learning,” Microsoft Learn, Apr. 2023. [Online]. Disponible a: <https://learn.microsoft.com/es-es/azure/machine-learning/how-to-understand-automated-ml?view=azureml-api-2>
- [50] “Classification (Keras) - Edge Impulse Documentation.” . [Online]. Disponible a: <https://docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks/classification>

- [51] Alvaro, “¿Qué es el sobreajuste u overfitting y por qué debemos evitarlo?,” MachineLearningParaTodos.com, Maig 2020, [Online]. Disponible a: <https://machinelearningparatodos.com/que-es-el-sobreajuste-u-overfitting-y-por-que-debemos-evitarlo/>
- [52] “Inteligencia artificial fácil - Machine Learning y Deep Learning prácticos - Funciones de pérdida (Loss function) | Ediciones ENI.” [Online]. Disponible a: <https://www.ediciones-eni.com/open/mediabook.aspx?idR=8dd2ca32769cb24b49648b15ef8e777e>
- [53] “Análisis de regresión—ArcGIS Insights | Documentación.” 2023, [Online]. Disponible a: <https://doc.arcgis.com/es/insights/latest/analyze/regression-analysis.htm#:~:text=El%20an%C3%A1lisis%20de%20regresi%C3%B3n%20es, valores%20bas%C3%A1ndose%20en%20el%20modelo>
- [54] “Support Vector Machine (SVM),” MATLAB & Simulink. [Online]. Disponible a: <https://es.mathworks.com/discovery/support-vector-machine.html>
- [55] R. Cañadas, “Redes neuronales recurrentes | Qué son las RNN,” abdatum, Nov 2022. [Online]. Disponible a: <https://abdatum.com/tecnologia/redes-neuronales-recurrentes>
- [56] GAMCO, SL, “Qué es Red Neuronal Recurrente Concepto y definición. Glosario,” [Online]. Disponible a: GAMCO, SL. <https://gamco.es/glosario/red-neuronal-recurrente/>
- [57] Itelligent, “Machine Learning & NPL: cómo funciona un clasificador de documentos,” INTELLIGENT INFORMATION TECHNOLOGIES, Apr. 2019. [Online]. Disponible a: <https://intelligent.es/es/pln-clasificacion-automatizada/documentos/#:~:text=La%20extracci%C3%B3n%20de%20caracter%C3%ADsticas%20>
- [58] S.A. Castaño, “Comunicación I2C,” Control Automático Educación, Jan. 2022, [Online]. Disponible a: <https://controlautomaticoeducacion.com/microcontroladores-pic/comunicacion-i2c/>
- [59] Jecrespom, “I2C,” Aprendiendo Arduino, Sep. 2017. [Online]. Disponible a: [https://aprendiendoarduino.wordpress.com/2017/07/09/i2c/#:~:text=La%20principal%20caracter%C3%ADstica%20de%20I%C2%B2C,es%20la%20referencia%20\(masa\)](https://aprendiendoarduino.wordpress.com/2017/07/09/i2c/#:~:text=La%20principal%20caracter%C3%ADstica%20de%20I%C2%B2C,es%20la%20referencia%20(masa))

- [60] “3 razones para aprender a programar en Python | KeepCoding,” KeepCoding Bootcamps, May 30, 2023. [Online]. Disponible a: <https://keepcoding.io/blog/razones-aprender-programar-python/>
- [61] “Scikit-Learn: Descubre la biblioteca de Python dedicada al Machine Learning,” Formation Data Science | DataScientest.com, Sep. 2022. [Online]. Disponible a: <https://datascientest.com/es/scikit-learn-decubre-la-biblioteca-python>
- [62] N. Palou, “Edge AI: Inteligencia Artificial fuera de la nube - Think Big Empresas,” Think Big, Abr. 2023. [Online]. Disponible a: <https://empresas.blogthinkbig.com/que-es-edge-ai-inteligencia-artificial-ventajas-aplicaciones/#:~:text=Edge%20AI%20se%20refiere%20a,de%20un%20servidor%20en%20internet>