

Trabajo Fin de Grado

Caracterización de baterías a partir de datos
abiertos mediante técnicas de inteligencia
artificial

Battery Characterization on the Basis of Open
Data Using Artificial Intelligence Techniques

Autor

Aida Martin Lacosta

Director

Tomas Cortes Arcos

Escuela Universitaria Politécnica La Almunia

Junio 2022



**Escuela Universitaria
Politécnica** - La Almunia
Centro adscrito
Universidad Zaragoza

**ESCUELA UNIVERSITARIA POLITÉCNICA
DE LA ALMUNIA DE DOÑA GODINA (ZARAGOZA)**

MEMORIA

Caracterización de baterías a partir de datos
abiertos mediante técnicas de inteligencia
artificial

Battery Characterization on the Basis Of Open
Data Using Artificial Intelligence Techniques

425.21.86

Autor: Aida Martin LacostaAida Martin Lacosta

Director: Tomas Cortes Arcos

Fecha: 21 06 2022



INDICE DE CONTENIDO BREVE

1. RESUMEN	5
2. ABSTRACT	6
3. ESTADO DEL ARTE	7
4. DESARROLLO	19
5. RESULTADOS	24
6. CONCLUSIONES	30
7. OBJETIVOS DE DESARROLLO SOSTENIBLE	31
8. BIBLIOGRAFÍA	32

INDICE DE CONTENIDO

1. RESUMEN	5
1.1. PALABRAS CLAVE	5
2. ABSTRACT	6
2.1. KEY WORDS	6
3. ESTADO DEL ARTE	7
3.1. ANTECEDENTES	7
3.1.1. <i>Klass et al. – 2014 [1]</i>	7
3.1.2. <i>Ma et al. – 2019 [2]</i>	7
3.1.3. <i>Uddin et al. – 2017 [3]</i>	8
3.1.4. <i>Zahid et al. – 2018 [4]</i>	8
3.1.5. <i>He et al. – 2014 [5]</i>	9
3.2. MARCO TEÓRICO	10
3.2.1. <i>Inteligencia Artificial [6]</i>	10
3.2.2. <i>Aprendizaje Automático</i>	11
3.2.2.1. Modelos Supervisados	11
3.2.2.1.1. Clasificación	11
3.2.2.1.2. Regresión	14
3.2.2.2. Modelos No Supervisados	16
3.2.3. <i>Baterías</i>	16
3.2.3.1. Batería Li-Ion	16
3.2.3.2. IRN 18650-20R	17
3.2.3.3. A123	17
4. DESARROLLO	19
4.1. SELECCIÓN DEL MODELO	19
4.2. DISEÑO DEL PROGRAMA	19
4.2.1. <i>Datos</i>	20
4.2.2. <i>Librerías</i>	20
4.2.3. <i>Programa Baterías</i>	20
4.2.4. <i>Programa Tensión</i>	22
5. RESULTADOS	24



Caracterización de baterías a partir de datos abiertos mediante técnicas de inteligencia artificial

Resumen

5.1. MODELO	24
5.1.1. <i>Clasificación</i>	24
5.1.2. <i>Regresión</i>	25
5.2. CLASIFICACIÓN DE BATERÍAS	27
5.3. PREDICCIÓN DE TENSIÓN	28
6. CONCLUSIONES	30
6.1. TRABAJOS FUTUROS	30
7. OBJETIVOS DE DESARROLLO SOSTENIBLE	31
8. BIBLIOGRAFÍA	32

INDICE DE ILUSTRACIONES

Ilustración 1. Ejemplo del funcionamiento de KNN.	12
Ilustración 2. Ejemplo de SVM [11].....	13
Ilustración 3. Ejemplos de hiperplano. [11].....	13
Ilustración 4. Ejemplo de regresión lineal [13]	14
Ilustración 5. Esquema de Random Forest Regressor [15]	15
Ilustración 6. Esquema de funcionamiento de las celdas Ion-Litio [17]	16
Ilustración 7. Batería IRN 18650-20R [18].....	17
Ilustración 8. Batería A123. [18]	18
Ilustración 9. Diagrama UML Programa Baterías	21
Ilustración 10. Esquema del proceso de programación con PyCaret.....	21
Ilustración 11. Diagrama UML Programa Tensión	22
Ilustración 12. Matriz de confusión clasificador	27
Ilustración 13. Error de Predicción por Clases.....	28
Ilustración 14. Error de Predicción de la regresión.	28

INDICE DE TABLAS

Tabla 1. Resultados compare_models(). Clasificación de Baterías.	24
Tabla 2. Resultados compare_models(). Predicción de tensión.....	25
Tabla 3. Resultados final del entrenamiento de la clasificación.....	27
Tabla 4. Resultados finales del entrenamiento de la regresión.	28

1. RESUMEN

En este trabajo se va a realizar la implementación técnicas de inteligencia artificial para la clasificación y estimación de características de baterías a partir de datos abiertos.

Para llevar a cabo esta clasificación y estimación se van a realizar dos algoritmos de aprendizaje automático, uno basado en la regresión y otro en la clasificación.

El modelo de regresión será utilizado para la predicción de la tensión en un punto determinado del tiempo de una celda de batería. Para ello, se entrenara la red neuronal con un dataset de medidas de un tipo de batería, en este caso, la celda IRN 18650-20R

El modelo de clasificación será utilizado para la clasificación de distintos tipos de baterías según sus datos característicos. Para ello, este modelo se ha entrenado con distintos dataset de distintas baterías.

Para llevar a cabo el trabajo, se ha empleado las librerías NumPy, Pandas, PyCaret y Matplotlib.

1.1. PALABRAS CLAVE

Aprendizaje automático, baterías, clasificación, regresión, inteligencia artificial.

2. ABSTRACT

In this work, the implementation of artificial intelligence techniques will be used for the classification and the estimation of battery characteristics from open data.

To do this classification and estimation, two machine learning algorithms will be done, one based on regression and the other one based on classification.

The regression model will be used for the prediction of the voltage at a given point of time of a battery cell. This will be done by a neuronal network trained with a dataset of measurements of a type of data, in this case, the IRN 18650-20R cell.

The classification model will be used for the classification of different types of batteries according to their characteristic data. This will be done by a model trained with different datasets from different batteries.

In this work, different libraries has been used. This libraries are NumPy, Pandas, PyCaret and Matplotlib.

2.1. KEY WORDS

Machine learning, batteries, classification, regression, artificial intelligence.

3. ESTADO DEL ARTE

3.1. ANTECEDENTES

En este apartado se van a proceder a explicar los distintos papers consultados para la realización de este proyecto de fin de grado.

3.1.1. Klass et al. – 2014 [1]

En este paper, se busca un método de estimación basado en SVM del estado de salud de baterías de Li-Ion en funcionamiento en vehículos eléctricos.

Durante la aplicación de baterías a bordo en vehículos eléctricos, los parámetros esenciales que se quieren monitorizar son la capacidad y la resistencia. Ya que los test realizados en laboratorios para la determinación del estado de la salud de la batería no se pueden realizar en el coche, este paper desarrolla un método donde estos test pueden ser realizados virtualmente a un modelo de batería basado en SVM. Este está basado en variables como la corriente, el voltaje y la temperatura, datos disponibles durante el uso del vehículo eléctrico.

En este paper se introducen nuevas características, como podrían ser la estimación de la capacidad y la dependencia de la temperatura, además de realizar una valoración del método explicado en el párrafo anterior. Para estimar la capacidad y resistencia, se generan y se utilizan los datos de uso de baterías de vehículos eléctricos típicos. Esta estimación se compara con mediciones hechas directamente a los indicadores de estado de salud, estas mediciones se han obtenido con tests estándar.

Después de la comparación, en el paper estima que estas estimaciones tienen una buena precisión en un rango de temperatura y estado de carga, lo cual permite la detección de la degradación. Además, en el artículo, se destaca que el método propuesto es adecuado para la aplicación en vehículos eléctricos.

3.1.2. Ma et al. – 2019 [2]

En este paper, se busca un nuevo índice de salud para el modelo de degradación RUL (vida útil restante) de la batería y los pronósticos.

Las baterías de ion de litio son usadas diariamente, pero el uso, la actuación de estas decrece debido al cambio de las propiedades físicas internas. El resultado que más resalta a primera vista de esta

degradación, es la reducción de la capacidad. Por eso, la predicción efectiva de la vida útil restante es importante.

En este papel, son introducidos dos nuevos indicadores de saludo: diferencia de tiempo de descarga del intervalo de igual voltaje y diferencia de temperatura de descarga del mismo intervalo de tiempo. Estos índices representan el proceso de degradación de las baterías de ion de litio. Para analizar la relación entre estos índices y la capacidad, se usa el coeficiente de correlación de Pearson y posteriormente, se usa una regresión de soporte vectorial para establecer el modelo de regresión de la vida útil restante.

Por último, en este paper se valida el método propuesto mediante el análisis de un dataset de baterías de litio de la nasa.

3.1.3. Uddin et al. – 2017 [3]

En este paper, se estudia la posibilidad de extender la vida útil de las baterías de ion de litio a través de V2G facilitado por un vehículo y un sistema de red inteligente.

En estos años, ha empezado a existir una necesidad de almacenamiento de energía. Esto se debe a que las energías renovables se han vuelto un pilar base en el sector eléctrico, pero debido a la variabilidad y la incertidumbre de estas, crea esta necesidad de almacenar energía.

En Uddin et al. se propone una solución a este problema. Esta solución es usar las baterías equipadas en vehículos eléctricos para facilitar la regulación de la frecuencia y el equilibrio de carga a través de tecnologías de vehículo a red (V2G, vehicle to grid).

Para ahondar en la variabilidad de las tecnologías V2G, se desarrolla un modelo integral de degradación de la batería basada en los datos de envejecimiento recopilados durante más de cincuenta experimentos de degradación a largo plazo en baterías comerciales $C_6/LiNiCoAlO_2$. Este modelo tiene en cuenta todos los modos de degradación existentes, como la edad de calendario, la capacidad de rendimiento, la temperatura, el estado de carga, la tasa de corriente y la profundidad de descarga.

Este modelo es validado en el paper. Una vez validado, este es integrado en una red inteligente diseñada para minimizar la degradación de la batería. Esta red es testeada en el paper, dando muy buenos resultados.

3.1.4. Zahid et al. – 2018 [4]

En este paper, se explica la estimación del estado de carga para la batería de un vehículo eléctrico usando un algoritmo de aprendizaje automático bajo ciclos de conducción diversificados.

La estimación del estado de carga es uno de los factores para resolver los problemas de monitoreo y seguridad de una batería de un vehículo eléctrico.

En este paper, es presentado y evaluado un nuevo enfoque para la estimación del estado de carga. Se usa un sistema neuro-difuso basado en el agrupamiento sustractivo para la estimación. Mientras que para la evaluación es usado un simulador avanzado de vehículo.

Los parámetros usados para modelar la estimación son los siguientes: la corriente, la temperatura, la pérdida de energía actual, potencia disponible y potencia solicitada, temperatura del aire del sistema de enfriamiento y un factor térmico de la batería.

Se utilizan datos recopilados de 10 ciclos distintos de conducción, estos datos son usados para el entrenamiento y el testeo de las etapas del modelo de estimación del estado de carga.

Como final del paper, se muestran los resultados de la evaluación, llegando a la conclusión del que modelo muestra un avance notable en la estimación del estado de carga respecto a anteriores estimaciones de este estado. Además, también es realizado un análisis de sensibilidad para determinar la importancia de cada parámetro dentro del modelo.

3.1.5. He et al. – 2014 [5]

En este paper, se explica la estimación del estado de carga para baterías de ion de litio usando un modelo de red neuronal y cancelación de error basado en un filtro Kalman.

Las baterías de ion litio se utilizan diariamente para almacenamiento de energía en dispositivos electrónicos, vehículos, aplicaciones aeroespaciales y sistemas de telecomunicaciones.

Existen varios métodos para estimar y gestionar el estado de carga de la batería para evitar la pérdida repentina de energía en los dispositivos previamente mencionados.

En este paper, se desarrolla un modelo de batería basado en una red neuronal artificial para la estimación del estado de carga de la batería en función de la corriente y el voltaje medidos. Se utiliza un filtro Kalman para reducir los errores en la estimación del estado de carga.

Este método se valida dentro del paper usando datos de la batería LiFePO₄ de 'Federal Driving Schedule' y pruebas de estrés dinámico.

3.2. MARCO TEÓRICO

A continuación se procederá a explicar los principios teóricos empleados en este trabajo, además de una breve explicación de los tipos de celdas usados en este trabajo.

3.2.1. *Inteligencia Artificial [6]*

La inteligencia artificial es un campo que ha aparecido muy recientemente en el campo de la ingeniería y ciencia, habiendo aparecido por primera vez en 1956. Aunque actualmente contiene una gran variedad de subcampos (desde aprendizaje y percepción, a inteligencias artificiales capaces de jugar a video juegos o realizar tareas cotidianas del día a día).

Esta área de la ingeniería no posee una definición concreta, sino que posee múltiples definiciones dependiendo del enfoque de cada persona. Los cuatro principales enfoques son los siguientes:

- **Thinking Humanly** o pensar humanamente: Este enfoque está basado en el modelado cognitivo. Pensar humanamente en Inteligencia Artificial se podría definir como hacer a la maquina "pensar", dotar a esta de actividades asociadas al pensamiento humano, como podría ser, resolución de problemas, aprendizaje, etc.
- **Thinking Rationally** o pensar racionalmente: Este enfoque está basado en las "leyes del pensamiento". Esta vertiente deriva del uso del campo de la lógica, iniciada por Aristóteles y su intento de codificar el pensamiento en base a silogismos. Pensar racionalmente en Inteligencia artificial podría definirse como el estudio de las facultades mentales (percepción, razón y actuación) usando modelos computacionales.
- **Acting Humanly** o actuar humanamente: Este enfoque está basado en el test de Turing. El test de Turing consiste en hacer pasar a un sistema computacional por un test realizado por un humano (interrogador), si el interrogador no puede decir si se trata de una maquina o un humano, el sistema pasa la prueba. Para pasar la prueba la maquina debería poseer las siguientes capacidades: procesamiento de lenguaje natural, representación del conocimiento, razonamiento automático y aprendizaje automático. Por tanto, actuar humanamente en Inteligencia Artificial podría definirse como la creación de maquinas para realizar tareas que requieren inteligencia para ser realizadas por personas.

- Acting Rationally o actuar racionalmente. Este enfoque está basado en el agente racional. En el contexto de este termino, un agente es un sistema que opera autónomamente, percibe su entorno, se adapta y cambia. Un agente racional se encarga de buscar la mejor salida.

Las bases de todos estos enfoques son las mismas: filosofía, matemáticas, economía, neurociencia, psicología, informática, lingüística, teoría del control y cibernética.

3.2.2. Aprendizaje Automático

Según Tom Mitchell, el aprendizaje automático o machine learning se podría definir como un programa que aprende de la experiencia con respecto a unas clases de variables y medidas del rendimiento (P), si el rendimiento en las variables, medido por P, mejora la experiencia.

Hay distintos tipos de aprendizaje automático, dependiendo de la tarea que va a realizar, la evaluación del sistema y la naturaleza de su entrenamiento. [7]

3.2.2.1. Modelos Supervisados

Es el tipo más usado en el área del aprendizaje automático. En este modelo, los datos introducidos a la hora de entrenar el sistema están compuestos por datos etiquetados (x (input), y (output)).

Dentro de esta categoría se encuentran los dos siguientes tipos de modelos supervisados: clasificación y regresión.

3.2.2.1.1. Clasificación

En este tipo de método, la variable Y, output o target, es una etiqueta conocida como clase. A la clasificación también se le conoce como reconocimiento de patrones. En este caso, el modelo predice la clase dados unos puntos x, clasificando los datos por categorías. [8]

Hay muchos tipos de clasificadores pero en este trabajo nos vamos a centrar en los más usados comúnmente: k-NN (k-Nearest Neighbor) y SVM (Support Vector Machine). [8]

3.2.2.1.1.1. K-Nearest Neighbors

El clasificador KNN asume que los nuevos puntos tienen categorías y características similares a su vecino más cercano. KNN se basa en la idea de calcular la distancia entre puntos en un gráfico, es decir, la distancia entre el punto nuevo y los puntos a su alrededor. El algoritmo

asignara las características y etiqueta del punto que más cerca este a este nuevo punto. [9]

El algoritmo funciona de la siguiente manera:

- Se selecciona el numero K de vecinos con los que se quiere comparar el nuevo punto.
- Calcula la distancia euclidiana entre el punto y los vecinos.
- Coge los K vecinos más cercanos al punto, para eso utiliza la distancia euclidiana calculada previamente.
- Entre los puntos seleccionados se coge la etiqueta de estos.
- Se asigna la etiqueta que es mas comun entre los vecinos al nuevo punto.

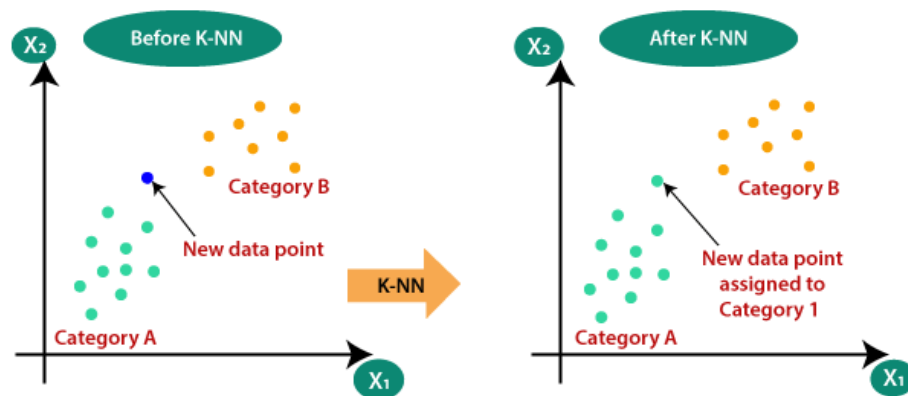


Ilustración 1. Ejemplo del funcionamiento de KNN.

A la hora de seleccionar la K, hay que tener en cuenta los puntos expuestos a continuación:

- No hay una forma particular para calcular el mejor valor de K. Se trata de un procedimiento de varias pruebas hasta encontrar el valor que mejor funcione para el modelo que es usado.
- Los valores de K presentan mucho ruido. [10]

3.2.2.1.1.2. Support Vector Machine

El objetivo del algoritmo de SVM o Support Vector Machine es el de encontrar un hiperplano en un espacio con N dimensiones que clasifica los puntos.

La regresión es un método estadístico para ayudar a comprender la relación entre variables dependientes (y) e independientes (x). Este tipo de aprendizaje automático predice valores continuos o reales.

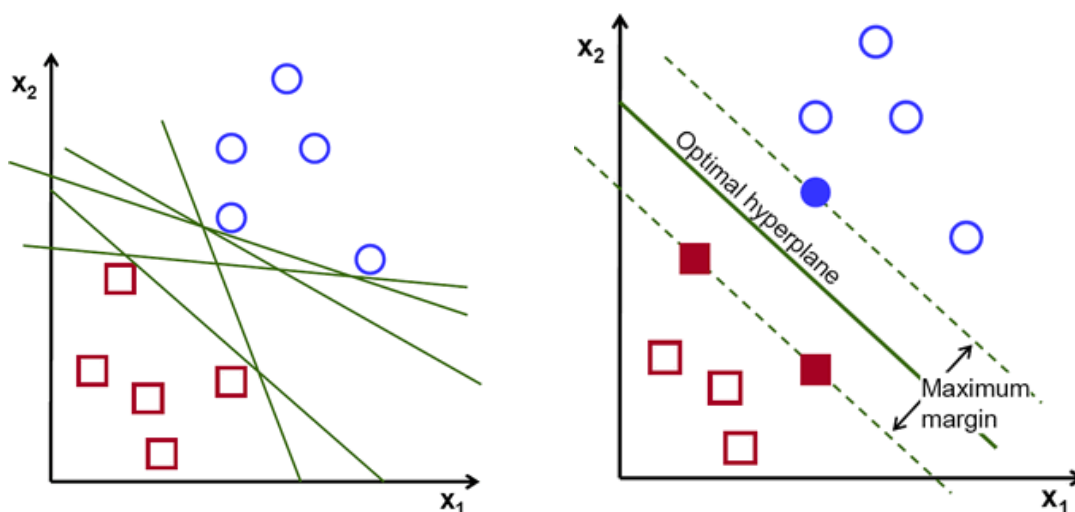
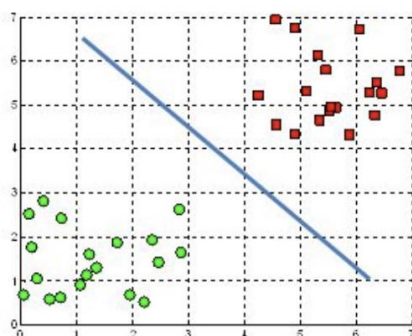


Ilustración 2. Ejemplo de SVM [11]

El objetivo de este algoritmo es encontrar un plano que maximiza el margen (máxima distancia entre los puntos de dos clases). Para separar las dos clases de datos, pueden usarse múltiples hiperplanos.

Los hiperplanos son límites de decisión que ayudan a separar los puntos de datos, que podrían representar a distintas clases. La dimensión del hiperplano depende del número de inputs o características. Un ejemplo sería el que de dos dimensiones se obtiene una línea o que de tres dimensiones se obtiene un plano de dos dimensiones. [11]

A hyperplane in \mathbb{R}^2 is a line



A hyperplane in \mathbb{R}^3 is a plane

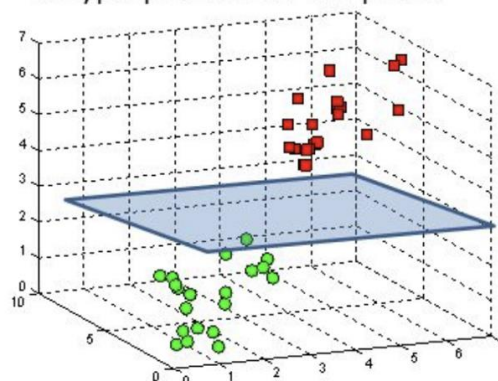


Ilustración 3. Ejemplos de hiperplano. [11]

Para poder manipular en la posición y orientación del hiperplano se usan vectores soporte, maximizando el margen del clasificador. Los vectores soporte son los puntos más cercanos al hiperplanos. Modificar o eliminar hiperplanos cambiara la posición y orientación de este. El SVM se compone de estos vectores soporte. [11]

3.2.2.1.2. Regresión

En este método, la variable Y ya no es una etiqueta o clase, sino que es un valor real, es decir, son valores continuos, a diferencia de la clasificación, que se apoya en la utilización de valores discretos. [12]

Al igual que los clasificadores, hay distintos tipos de regresiones, pero en este trabajo nos vamos a centrar en uno de los más usados, además de explicar el modelo usado en este proyecto: regresión lineal y Random Forest Regressor.

3.2.2.1.2.1. Regresión lineal

La regresión lineal es la regresión en la que las variables tienen una relación lineal entre ellas.

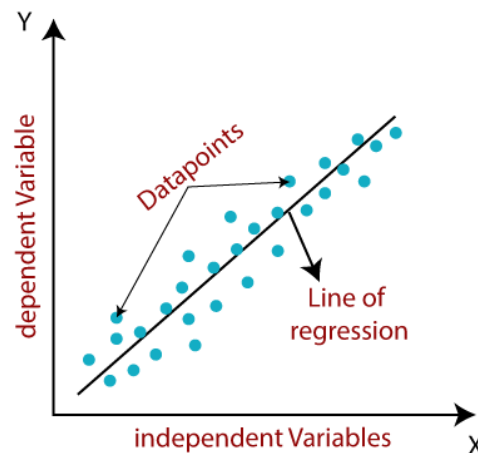


Ilustración 4. Ejemplo de regresión lineal [13]

Este modelo está basado en la ecuación lineal:

$$y = a_0 + a_1 * x$$

Donde:

- Y es la variable dependiente
- X es el variable independiente
- a_0 y a_1 son los parámetros lineales a calcular.

En este algoritmo, la misión principal es encontrar los mejores valores de los parámetros lineales para ajustar la línea de regresión con el menor error posible. El método usado para este cálculo es la función de coste. La función de coste optimiza estos parámetros. En la regresión lineal se usa la función de coste MSE (Mean Squared Error), la cual está representada por la siguiente ecuación:

$$MSE = \frac{1}{n} \sum_{i=1}^n (pred_i - y_i)^2$$

Donde:

- N es el número de observaciones
- y_i es el valor actual de y
- $pred_i$ es el valor predicho

Para minimizar el error, se calcula el gradiente descendiente de la función de coste MSE. Este gradiente también se usa para actualizar los coeficientes y reducir su función de coste. [13]

3.2.2.1.2.2. Random Forest Regressor

Este algoritmo de aprendizaje supervisado está basado en el uso del aprendizaje conjunto. Este tipo de aprendizaje une predicciones de distintos algoritmos de aprendizaje automático para conseguir la predicción más acertada. [14], [15]

La estructura de este modelo es la siguiente:

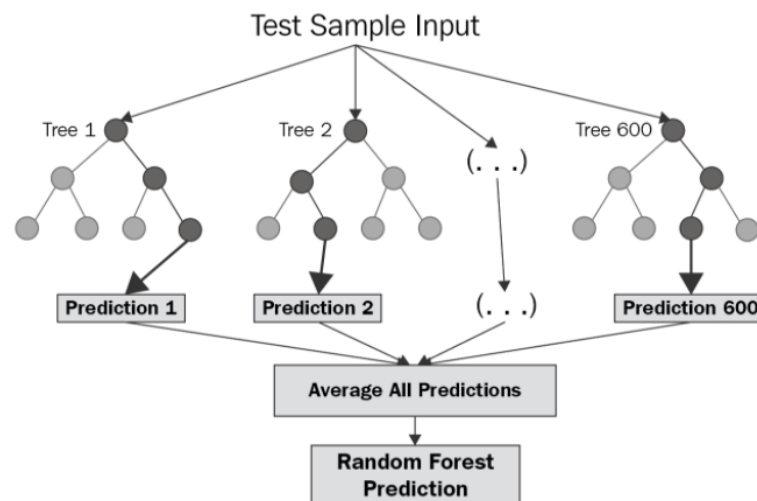


Ilustración 5. Esquema de Random Forest Regressor [15]

Los pasos que realiza este algoritmo serían los siguientes:

- Seleccionar un número aleatorio de puntos para el entrenamiento del modelo.
- Construir un árbol de decisiones
- Elegir un número N de árboles para conseguir distintas predicciones. Se debe repetir los dos pasos anteriormente expuestos.

- Cada árbol de decisiones calcula una predicción y. Una vez tenemos todas las predicciones de los árboles, se calcula la media de estas y se asocia este valor y al valor o valores x introducidos en el algoritmo.

3.2.2.2. Modelos No Supervisados

En este tipo de modelos, el set de datos no tiene Y ni targets. Estos son utilizados para interpretar datos, en lugar de "mapearlos" como realiza el modelo supervisado. Es el modelo más aproximado a la forma en la que los seres humanos procesan la información. [7]

3.2.3. Baterías

En esta sección del trabajo, se va a proceder a explicar el funcionamiento de los dos tipos de celdas usados en este proyecto. Pero antes, se va a hacer una breve introducción a las baterías de ion de Litio.

3.2.3.1. Batería Li-Ion

Aunque las baterías de Ion-Litio tienen muchas variables, todos tienen en común su funcionamiento: un cátodo, un ánodo y un conductor, en este caso, electrolítico. [16]

Toda la electroquímica de estas baterías está basada en los iones litio, como el propio nombre de esta tecnología indica. Durante la descarga, los iones de litio pasan del ánodo hasta el cátodo, pasando por el electrolito. En el cátodo, el ion se recombina con otros iones y se neutralizan eléctricamente. Este proceso se revierte durante la carga, el ion viaja del ánodo al cátodo. El cátodo y el ánodo están separados por una fina membrana, la cual solo permite el paso de los iones de Litio. [17]

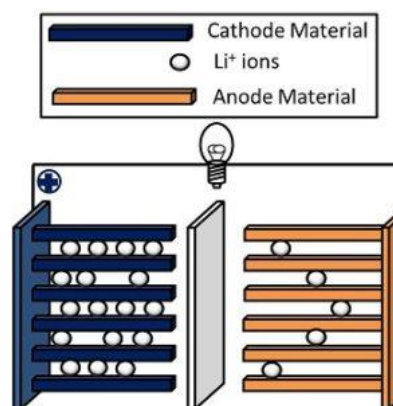


Ilustración 6. Esquema de funcionamiento de las celdas Ion-Litio [17]

3.2.3.2. IRN 18650-20R

Este tipo de celdas cilíndricas de Li-Ion no son muy grandes ni muy pesadas, tienen un tamaño de aproximadamente 18.33 mm de diámetro y 64.85 mm de largo y un peso de 45 g, sin incluir el circuito de seguridad. Su capacidad es de 2000 mAh. [18]

La química de estas baterías se basa en LNMC (LiNiMnCoO_2). Su cátodo es una combinación de Nickel, Magnesio y Cobalto. Este cátodo es de los más usados y una de las fórmulas más exitosas desarrolladas. [19]

El secreto de esta celda, reside en la unión de nickel y magnesio. La combinación de estos elementos permite mejorar las debilidades de cada uno, un ejemplo de esto sería la energía específica (la energía específica es la energía por unidad de masa), mientras que el nickel presenta una alta energía, el magnesio tiene una baja energía. [20]

Estas tienen un coste bajo, una densidad energética alta y un ciclo de vida largo.

Este tipo de batería tiene diversas aplicaciones, como puede ser, proporcionar energía a herramientas, bicicletas eléctricas, etc.



Ilustración 7. Batería IRN 18650-20R [18]

3.2.3.3. A123

Este tipo de celda de Li-Ion presenta un tamaño parecido a las explicadas en el apartado anterior, tiene un diámetro de 25.4 mm y 65 mm aproximadamente. Por el contrario, pesa algo más, el peso de la batería A123 es de 76 g. La capacidad de este tipo de celdas es de 2230 mAh. [18]

La química de estas baterías se basa en LiFePO_4 , litio-ferrofostato o fosfato de hierro-litio. Su cátodo es una combinación está compuesto por fosfato de hierro. [21]

Estas baterías estas ofrecen una baja resistencia, un ciclo de vida largo, además de tener estabilidad térmica, mayor seguridad y tolerancia al ser maltratadas. [20]

Las principales aplicaciones de estas celdas con industriales, que requieren altas corrientes y resistencia. [22]

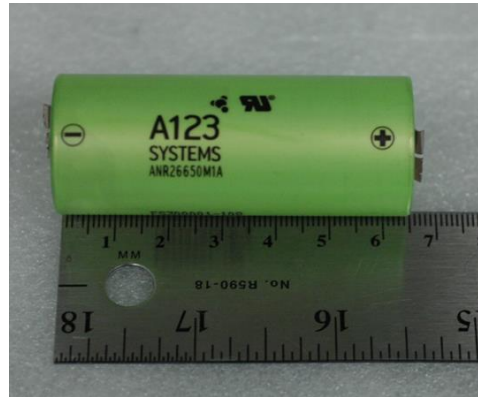


Ilustración 8. Batería A123. [18]

4. DESARROLLO

A continuación, en este apartado se va a proceder a explicar el desarrollo que se ha llevado a cabo durante el proyecto, desde la selección del modelo hasta el diseño de los programas y elección de las librerías.

4.1. SELECCIÓN DEL MODELO

A la hora de seleccionar el modelo, se ha tenido en cuenta los datos (tipo, distribución, tamaño, etc.) y la función que debe realizar la red neuronal o modelo.

Para el primer programa, clasificación de baterías, la función que realiza el modelo es la de clasificar la batería dependiendo de los datos discretos dados a la red, por tanto, se empleará un modelo de clasificación. Además, dado que se van a introducir datos a la red neuronal, este modelo será supervisado.

Para el segundo programa, predicción de la tensión de una batería, la finalidad que realiza es distinta a la del primero, en este caso, la función será la de predecir la tensión en función de los datos dados. Para ello, se empleará un modelo de regresión supervisado, ya que, al igual que el anterior, el modelo recibe unos datos.

Una vez se ha definido la base de ambos modelos, se procederá a seleccionar el tipo de regresión o clasificación que mejor se adapta a los datos. Para realizar esta tarea, se usará el comando de la librería PyCaret llamado `compare_models()`, esta función compara distintos modelos, tanto para regresión como para clasificación, y devuelve datos para la evaluación del modelo.

Estos datos serán analizados en el apartado de resultados 5.1.

4.2. DISEÑO DEL PROGRAMA

En este apartado, se va a explicar la creación y diseño de los dos programas realizados en el trabajo.

Para el desarrollo de la programación se ha empleado el software Spyder (anaconda 3) [23]. Se ha elegido este programa por las herramientas que ofrece, como puede ser el explorador de variables o de plots.

4.2.1. Datos

Los datos usados en el programa han sido descargados de la página CALCE. [18] Estos datos son abiertos y provienen de distintos tipos de baterías. En este caso, se ha seleccionado dos tipos de baterías para la clasificación, A123 e IRN 18650-20R. En cambio, para el segundo modelo se ha usado únicamente la batería IRN.

Esta selección de datos se debe a la limitación del hardware empleado, para una mayor fluidez durante el trabajo.

4.2.2. Librerías

La elección de librerías para la realización de este proyecto ha seguido los siguientes criterios:

- Librería para el tratamiento de los datos
- Librería con funciones relacionadas con el aprendizaje automático
- Librería para el generado gráficos con los datos.

Siguiendo estos criterios, se han escogido las siguientes librerías:

- NumPy [24]: Librería para Python que permite manipular y crear vectores y matrices.
- Pandas [25]: Librería para Python para manipulación y análisis de datos.
- PyCaret [26]: Librería para Python de código abierto que permite desde preparar los datos hasta el guardado del modelo.
- Matplotlib [27]: Librería para Python para crear gráficos e imágenes a partir de datos.

Estas son las principales librerías que se han usado durante el proceso de programación.

4.2.3. Programa Baterías

Para la realización de este programa se han utilizado los ficheros los ficheros completos de IRN y A123. Se ha seleccionado dos ficheros de datos de IRN y uno de A123 para el entrenamiento, mientras que para el testeo se han utilizado un fichero de IRN y uno de A123 no usados previamente.

Para la explicación de la programación del modelo de clasificación, se ha realizado un diagrama UML expuesto en la siguiente ilustración.



Ilustración 9. Diagrama UML Programa Baterías

En este programa, mediante las librerías explicadas en el apartado 4.2.2, se va a realizar el tratamiento de datos, creación y entrenamiento del modelo de clasificación para baterías.

Para la lectura y concatenación de los ficheros de datos se han creado dos pequeñas funciones, una para la concatenación de las hojas del Excel y otra para la concatenación de las listas en DataFrames.

El set up, comparación, creación y evaluación del modelo se ha realizado con la librería PyCaret siguiendo el siguiente esquema [28]:

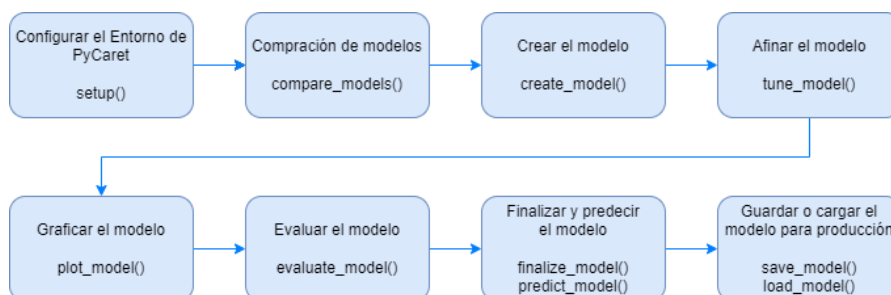


Ilustración 10. Esquema del proceso de programación con PyCaret

Después de utilizar el comando `compare_models()` se elegirá el mejor modelo de los estudiados y se realizarán el resto de pasos con este modelo.

Los resultados del modelo empleado y del resto de modelos, además de la justificación de la elección, están expuestos en el apartado 5.2.

El código completo se encuentra en los anexos de este documento en el apartado cuyo nombre es `Baterias.py`.

4.2.4. Programa Tensión

A diferencia del programa anterior, en este caso, se ha usado únicamente un fichero de la batería IRN: "11_05_2015_SP20-2_DST_50SOC.xls". En este fichero seleccionaremos una fila cada 3 posiciones para la prueba.

Para la explicación de la programación del modelo de clasificación, se ha realizado un diagrama UML expuesto en la siguiente ilustración.

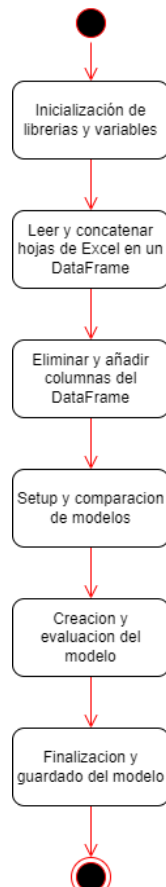


Ilustración 11. Diagrama UML Programa Tensión

En este programa, mediante las librerías explicadas en el apartado 4.2.2 se va a realizar el tratamiento de datos, creación y entrenamiento del modelo de regresión para la predicción de la tensión en un punto determinado del tiempo.

El proceso de creación del programa será el mismo exceptuando la selección y tratamiento de los datos para el entrenamiento y test. En esta ocasión, se cogerá uno de cada tres datos del DataFrame. Esto es debido a que se quiere realizar un modelo con buena predicción usando datos puntuales cada x tiempo.

El código completo se encuentra en los anexos de este documento en el apartado cuyo nombre es Tension.py.

5. RESULTADOS

A continuación, se exponen los resultados obtenidos en las distintas partes del trabajo.

5.1. MODELO

5.1.1. Clasificación

En un modelo de clasificación, como se ha explicado en el apartado 3.2.2.1.1, para que un modelo de clasificación sea bueno tiene que presentar un nivel alto de accuracy, de precisión y un buen recall.

Tabla 1. Resultados compare_models(). Clasificación de Baterías.

	Model	Accuracy	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
dt	Decision Tree Classifier	1.0	0.9999	1.0	1.0	0.9999	0.9999	0.01
rf	Random Forest Classifier	1.0	0.9999	1.0	1.0	0.9999	0.9999	0.178
ada	Ada Boost Classifier	1.0	0.9999	1.0	1.0	0.9999	0.9999	0.152
gbc	Gradient Boosting Classifier	1.0	0.9999	1.0	1.0	0.9999	0.9999	0.412
et	Extra Trees Classifier	1.0	1.0	1.0	1.0	1.0	1.0	0.089
xgboost	Extreme Gradient Boosting	1.0	0.9999	1.0	1.0	0.9999	0.9999	0.209
lightgbm	Light Gradient Boosting Machine	1.0	1.0	1.0	1.0	1.0	1.0	0.184
catboost	CatBoost Classifier	1.0	1.0	1.0	1.0	1.0	1.0	3.084
knn	K Neighbors Classifier	0.9997	1.0	0.9994	0.9997	0.9994	0.9994	0.254
lda	Linear Discriminant Analysis	0.999	1.0	0.9979	0.9989	0.998	0.998	0.009
lr	Logistic Regression	0.9987	0.9999	0.9974	0.9986	0.9974	0.9974	0.503

	Model	Accuracy	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
ridge	Ridge Classifier	0.9972	1.0	0.9943	0.9971	0.9945	0.9945	0.007
svm	SVM - Linear Kernel	0.9324	0.8878	0.9704	0.9263	0.8641	0.8683	0.108
nb	Naive Bayes	0.8112	0.66	0.9248	0.7702	0.6175	0.6429	0.009
qda	Quadratic Discriminant Analysis	0.5205	0.0	0.0	0.0	0.0	0.0	0.01

Coger un modelo que presenta un 1 de cada uno de los valores mencionados en el párrafo anterior puede suponer que nuestro modelo sufra de sobreajuste, en cambio, coger unos valores muy pequeños puede suponer que el modelo este subajustado. Por tanto, se debe elegir un punto medio entre estos valores.

Siguiendo este criterio y observando los resultados obtenidos en la Tabla 1 se ha escogido el modelo SVM o también conocido como Support Vector Machine o Máquina de vectores de soporte.

El modelo SVM proporciona unos valores altos pero a la vez son lo bastante bajos como para no producirse overfitting o sobreajustado.

5.1.2. Regresión

En un modelo de regresión, como se ha explicado en el apartado 3.2.2.1.2, para que un modelo sea bueno tiene que presentar un nivel alto de R2 y unos niveles bajos de MAE, MSE y RMSE.

Tabla 2. Resultados compare_models(). Predicción de tensión.

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
et	Extra Trees Regressor	0.0026	0.0002	0.0095	0.9973	0.0023	0.0008	0.043
catboost	CatBoost Regressor	0.0043	0.0003	0.0124	0.996	0.0029	0.0013	0.457
gbr	Gradient Boosting Regressor	0.0061	0.0004	0.014	0.9948	0.0033	0.0018	0.018
xgboost	Extreme Gradient Boosting	0.0049	0.0004	0.0143	0.9941	0.0034	0.0015	0.141
rf	Random Forest Regressor	0.0049	0.0005	0.0155	0.9935	0.0037	0.0015	0.055

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
dt	Decision Tree Regressor	0.0055	0.0006	0.0194	0.9912	0.0046	0.0016	0.005
lightgbm	Light Gradient Boosting Machine	0.0082	0.0006	0.0204	0.9911	0.0048	0.0024	0.051
br	Bayesian Ridge	0.0169	0.0009	0.0274	0.9854	0.0063	0.0048	0.004
lr	Linear Regression	0.0169	0.0009	0.0274	0.9854	0.0063	0.0048	0.004
ridge	Ridge Regression	0.0178	0.001	0.0288	0.9841	0.0066	0.0051	0.004
ada	AdaBoost Regressor	0.0367	0.0027	0.0511	0.9537	0.0109	0.01	0.018
knn	K Neighbors Regressor	0.0345	0.0041	0.0629	0.9291	0.0143	0.01	0.006
omp	Orthogonal Matching Pursuit	0.0799	0.0128	0.112	0.7789	0.0239	0.0219	0.004
lasso	Lasso Regression	0.0934	0.0181	0.1335	0.6855	0.028	0.0253	0.004
en	Elastic Net	0.0935	0.0181	0.1335	0.6855	0.028	0.0253	0.004
llar	Lasso Least Angle Regression	0.1684	0.0577	0.2392	-0.0095	0.0506	0.0454	0.004
lar	Least Angle Regression	0.1247	0.1752	0.1634	-16.347	0.0391	0.0353	0.005
huber	Huber Regressor	0.1644	0.1729	0.3352	-22.503	0.0813	0.0439	0.009
par	Passive Aggressive Regressor	0.9809	22.648	14.448	-391.055	0.3094	0.2667	0.004

Al igual que en el caso anterior, hay que tener en cuenta los valores mencionados anteriormente para elegir un modelo que no esté overfitted ni underfitted.

Siguiendo este criterio y observando los resultados obtenidos en la Tabla 2 se ha escogido el modelo RF o también conocido como Random Forest Regressor.

El modelo RF proporciona un valor alto para R2, pero a la vez, es un valor lo bastante bajo como para no producirse overfitting o sobreajustado. Los distintos errores presentan un valor bajo.

5.2. CLASIFICACIÓN DE BATERÍAS

Una vez realizado todo el proceso expuesto en el apartado 4.2.3, se han obtenido los siguientes resultados del entrenamiento de la red.

Tabla 3. Resultados final del entrenamiento de la clasificación.

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
SVM - Linear Kernel	0.9823	0.9824	0.9832	0.9798	0.9815	0.9646	0.9646

Como se puede observar en la Tabla 3 después de realizar el entrenamiento y el testeo de la red neuronal, obtenemos unos valores muy altos de accuracy, AUC, Recall y Precisión. Por tanto, el modelo estará clasificando correctamente.

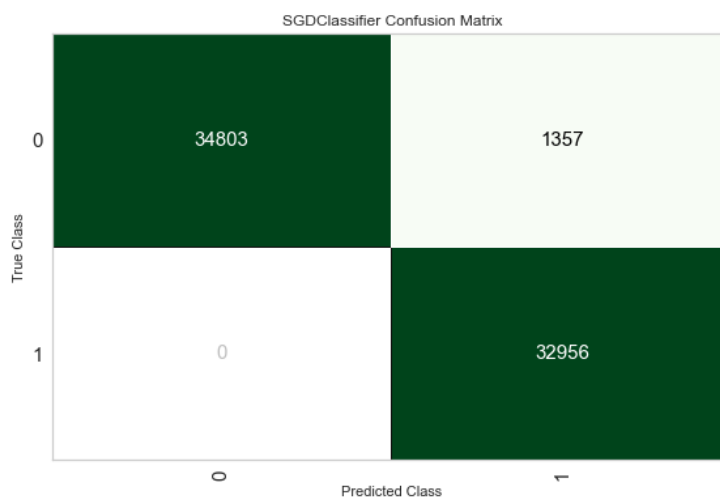


Ilustración 12. Matriz de confusión clasificador

En la Ilustración 12. Matriz de confusión clasificador, se puede visualizar la matriz de confusión del modelo, en la cual nos muestra las verdaderas categorías (filas) contra las categorías predichas (columnas). Esta matriz, en una situación en la que la red neuronal fuese perfecta, deberá presentar valores distintos de 0 solo en la diagonal principal. Como podemos apreciar, tenemos 1357 predicciones erróneas de las aproximadas 70.000 predicciones realizadas por la red.

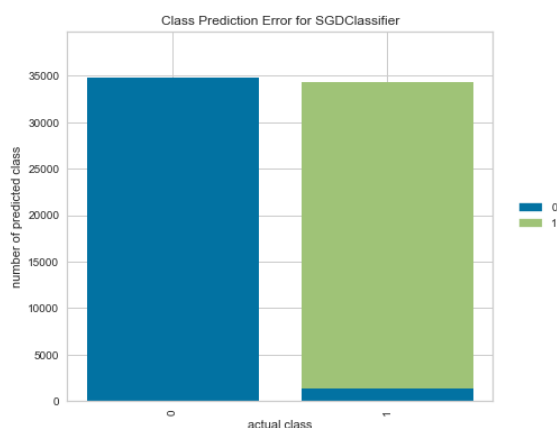


Ilustración 13. Error de Predicción por Clases

También se puede observar que todos los errores pertenecen a la predicción de una de las dos baterías. Aunque la cantidad de errores con respecto de los aciertos es despreciable.

5.3. PREDICCIÓN DE TENSIÓN

Una vez realizado todo el proceso expuesto en el apartado 4.2.4, se han obtenido los siguientes resultados del entrenamiento de la red.

Tabla 4. Resultados finales del entrenamiento de la regresión.

Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
Random Forest Regressor	0.0073	0.0002	0.0131	0.9972	0.003	0.0021

Como se puede observar en la Tabla 4 después de realizar el entrenamiento y el testeo de la red neuronal, obtenemos muy buenos resultados, con un bajo error y un gran nivel de predicción.

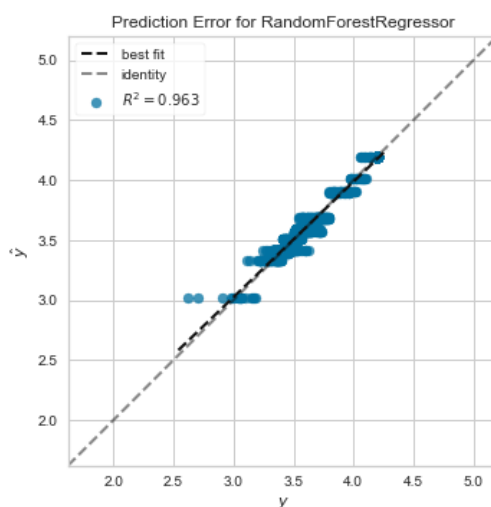


Ilustración 14. Error de Predicción de la regresión.



En la ilustración anterior, se puede observar que realiza un buen fit y de que los datos no presentan un error muy grande respecto al valor real de la predicción.

6. CONCLUSIONES

Después de todo lo analizado en el apartado anterior, se puede decir que se han cumplido los dos grandes objetivos de este trabajo: la obtención de dos modelos funcionales para la clasificación y predicción de valores con respecto a unos datos dados y la aplicación de técnicas de inteligencia artificial como puede ser el aprendizaje automático o las redes neuronales.

Como conclusiones finales, la manipulación de los datos y elección del modelo correcto puede influir en gran medida a la hora de generar una red neuronal o modelo consistente para desarrollar una función, como se puede observar en la comparación de los distintos modelos en ambos programas en el apartado 5.

6.1. TRABAJOS FUTUROS

Para trabajos futuros se plantea la implementación de un clasificador para más de dos celdas. Como se expone en el punto 4.2.1, fueron elegidas dos baterías por limitación de hardware, por tanto, se pueden añadir más tipos de celdas al clasificador.

El programa realizado cuenta con esta opción implementada y se ha añadido una pequeña leyenda para la identificación de los distintos tipos de celdas dentro del código.

En cuanto a la predicción de la tensión, este programa puede ser usado para distintos tipos de baterías y celdas. Este tipo de regresión, con unas modificaciones, se podría llegar a emplear para el cálculo del SOC (state of charge) o predecir cualquier otro tipo de variable.

7. OBJETIVOS DE DESARROLLO SOSTENIBLE

Los objetivos de este Trabajo Fin de Grado están alineados con los siguientes Objetivos de Desarrollo Sostenible (ODS) y metas, de la Agenda 2030:

- Objetivo 4 - Garantizar una educación inclusiva y equitativa de calidad y promover oportunidades de aprendizaje permanente para todos



- Meta 4.4 De aquí a 2030, aumentar considerablemente el número de jóvenes y adultos que tienen las competencias necesarias, en particular técnicas y profesionales, para acceder al empleo, el trabajo decente y el emprendimiento

- Objetivo 8 - Promover el crecimiento económico sostenido, inclusivo y sostenible, el empleo pleno y productivo y el trabajo decente para todos



- Meta 8.2 Lograr niveles más elevados de productividad económica mediante la diversificación, la modernización tecnológica y la innovación, entre otras cosas centrándose en los sectores con gran valor añadido y un uso intensivo de la mano de obra

8. BIBLIOGRAFÍA

- [1] V. Klass, M. Behm, y G. Lindbergh, «A support vector machine-based state-of-health estimation method for lithium-ion batteries under electric vehicle operation», *J. Power Sources*, vol. 270, pp. 262-272, dic. 2014, doi: 10.1016/j.jpowsour.2014.07.116.
- [2] «Ma et al. - 2019 - A Novel Health Index for Battery RUL Degradation M.pdf».
- [3] K. Uddin, T. Jackson, W. D. Widanage, G. Chouchelamane, P. A. Jennings, y J. Marco, «On the possibility of extending the lifetime of lithium-ion batteries through optimal V2G facilitated by an integrated vehicle and smart-grid system», *Energy*, vol. 133, pp. 710-722, ago. 2017, doi: 10.1016/j.energy.2017.04.116.
- [4] T. Zahid, K. Xu, W. Li, C. Li, y H. Li, «State of charge estimation for electric vehicle power battery using advanced machine learning algorithm under diversified drive cycles», *Energy*, vol. 162, pp. 871-882, nov. 2018, doi: 10.1016/j.energy.2018.08.071.
- [5] W. He, N. Williard, C. Chen, y M. Pecht, «State of charge estimation for Li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation», *Int. J. Electr. Power Energy Syst.*, vol. 62, pp. 783-791, nov. 2014, doi: 10.1016/j.ijepes.2014.04.059.
- [6] S. J. Russell, P. Norvig, y E. Davis, *Artificial intelligence: a modern approach*, 3rd ed. Upper Saddle River: Prentice Hall, 2010.
- [7] Kevin P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. [En línea]. Disponible en: probml.ai
- [8] S. Asiri, «Machine Learning Classifiers», *Medium*, 11 de junio de 2018. <https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623> (accedido 18 de noviembre de 2021).
- [9] O. Harrison, «Machine Learning Basics with the K-Nearest Neighbors Algorithm», *Medium*, 14 de julio de 2019. <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761> (accedido 18 de noviembre de 2021).
- [10] «K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint», *www.javatpoint.com*. <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning> (accedido 18 de noviembre de 2021).
- [11] R. Gandhi, «Support Vector Machine — Introduction to Machine Learning Algorithms», *Medium*, 5 de julio de 2018.

- <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>
(accedido 19 de noviembre de 2021).
- [12] Kevin P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. [En línea]. Disponible en: probml.ai
- [13] «Linear Regression in Machine learning - Javatpoint», *www.javatpoint.com*. <https://www.javatpoint.com/linear-regression-in-machine-learning> (accedido 19 de junio de 2022).
- [14] N. Beheshti, «Random Forest Regression», *Medium*, 2 de marzo de 2022. <https://towardsdatascience.com/random-forest-regression-5f605132d19d> (accedido 19 de junio de 2022).
- [15] C. Bakshi, «Random Forest Regression», *Medium*, 14 de abril de 2022. <https://levelup.gitconnected.com/random-forest-regression-209c0f354c84> (accedido 19 de junio de 2022).
- [16] «BU-204: How do Lithium Batteries Work?», *Battery University*, 22 de septiembre de 2010. <https://batteryuniversity.com/article/bu-204-how-do-lithium-batteries-work> (accedido 19 de junio de 2022).
- [17] «Lithium-Ion Battery», *Clean Energy Institute*. <https://www.cei.washington.edu/education/science-of-solar/battery-technology/> (accedido 19 de junio de 2022).
- [18] «CALCE Battery Group». <https://web.calce.umd.edu/batteries/data.htm> (accedido 18 de abril de 2021).
- [19] «NMC Battery Material for Li-ion Cells (LiNiMnCoO₂)», *Targray*. <https://www.targray.com/li-ion-battery/cathode-materials/nmc> (accedido 18 de junio de 2022).
- [20] «BU-205: Types of Lithium-ion», *Battery University*, 18 de septiembre de 2010. <https://batteryuniversity.com/article/bu-205-types-of-lithium-ion> (accedido 19 de junio de 2022).
- [21] «Baterías LiFePo₄ – Blog Baterías de litio». <https://www.bateriasdelitio.net/?p=104> (accedido 18 de junio de 2022).
- [22] «BU-216: Summary Table of Lithium-based Batteries», *Battery University*, 4 de septiembre de 2010. <https://batteryuniversity.com/article/bu-216-summary-table-of-lithium-based-batteries> (accedido 19 de junio de 2022).
- [23] «Home — Spyder IDE». <https://www.spyder-ide.org/> (accedido 16 de noviembre de 2021).
- [24] «NumPy». <https://numpy.org/> (accedido 16 de noviembre de 2021).

- [25] «pandas - Python Data Analysis Library». <https://pandas.pydata.org/> (accedido 16 de noviembre de 2021).
- [26] «Home», *PyCaret*. <https://pycaret.org/> (accedido 16 de noviembre de 2021).
- [27] «Matplotlib: Python plotting — Matplotlib 3.4.3 documentation». <https://matplotlib.org/stable/index.html> (accedido 16 de noviembre de 2021).
- [28] «4- Creando Todo El Pipeline De Machine Learning Con PyCaret y Python». <https://www.datasource.ai/uploads/95e92d244d76c691837d30957d93b1f4.html> (accedido 16 de noviembre de 2021).

Relación de documentos

(X) Memoria NN páginas

(_) Anexos NN páginas

La Almunia, a 21 de Junio de 2022



Firmado: Aida Martin Lacosta