



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**título del TFG
Documentación Técnica**



Presentado por José Luis Pérez Gómez
en Universidad de Burgos — 2 de julio de 2024
Tutor: Bruno Baruque

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	iv
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	2
A.3. Estudio de viabilidad	7
Apéndice B Especificación de Requisitos	9
B.1. Introducción	9
B.2. Objetivos generales	9
B.3. Catálogo de requisitos	10
B.4. Especificación de requisitos	12
Apéndice C Especificación de diseño	19
C.1. Introducción	19
C.2. Diseño de datos	21
C.3. Diseño procedimental	21
C.4. Diseño arquitectónico	23
Apéndice D Documentación técnica de programación	25
D.1. Introducción	25
D.2. Estructura de directorios	25
D.3. Manual del programador	26

D.4. Compilación, instalación y ejecución del proyecto	26
D.5. Pruebas del sistema	26
Apéndice E Documentación de usuario	27
E.1. Introducción	27
E.2. Requisitos de usuarios	27
E.3. Instalación	27
E.4. Manual del usuario	27
Apéndice F Anexo de sostenibilización curricular	29
F.1. Introducción	29

Índice de figuras

A.1. Commits realizados durante la realización del TFG	2
A.2. Additions realizados durante la realización del TFG	3
A.3. Deletions realizados durante la realización del TFG	3
A.4. Commits-Etapa 1	4
A.5. Commits-Etapa 2	5
A.6. Commits-Etapa 3	6
C.1. Cantidad de elementos para la característica Key	20
C.2. Análisis característica Timestamp del conjunto de datos	21

Índice de tablas

B.1. CU-001 Iniciar notebook principal (Main).	13
B.2. CU-002 Ejecución celda 1. Crear un entorno virtual.	13
B.3. CU-003 Ejecución celdas (Notebooks Preparatorios).	14
B.4. CU-004 Ejecución celdas cargar de archivo CSV.	15
B.5. CU-005 Ejecución celdas (Notebooks análisis y preprocessing) .	16
B.6. CU-006 Ejecución celdas (Notebooks experimentos)	17
B.7. CU-007 Ejecución celdas (Notebook resultados)	18

Apéndice A

Plan de Proyecto Software

A.1. Introducción

La Universidad de Burgos, dentro del área de conocimiento de Ingeniería de Sistemas y Automática, dispone de un interfaz BCI (Brain Computer Interface) para la captación de señales cerebrales. Empleando ese interfaz se han realizado diferentes experimentos que han permitido recoger información de la actividad cerebral mientras los usuarios ejecutaban diferentes tareas cotidianas.

Este Trabajo de Fin de Grado (TFG) tiene como objetivo el análisis de la información obtenida en esos experimentos. Se entrenarán diferentes algoritmos para clasificar la acción realizada por el usuario a partir de las señales generadas por el BCI. Con este propósito, se evaluarán diferentes algoritmos de procesamiento de señales y de machine/deep learning para la clasificación automática de señales.

Los datos aportados son de tipo EEG (Electroencefalografía) para la realización del TFG son datos referentes a experimentos basados en pulsaciones sobre teclas de un teclado: arriba, abajo, izquierda, derecha.

El análisis de estos datos y su evaluación en diferentes algoritmos esta basada en predecir qué teclas del teclado se han pulsado según las señales captadas con la interfaz BCI.

Para esto no ha habido una planificación como tal registrada en Github, pero sí una progresión definida en los commits de código generados durante la composición del TFG.

A.2. Planificación temporal

En la reunión inicial con los tutores definimos utilizar Python para realizar el código y una serie de aprendizajes básicos para poder acometer el TFG sin problemas.

En las siguientes reuniones se definieron algoritmos y experimentos a realizar con los datos EGG aportados.

Desde el principio del proyecto debido a las circunstancias personales del alumno no se ha podido realizar metodología Scrum, pero sí se produjeron estas líneas temporales:

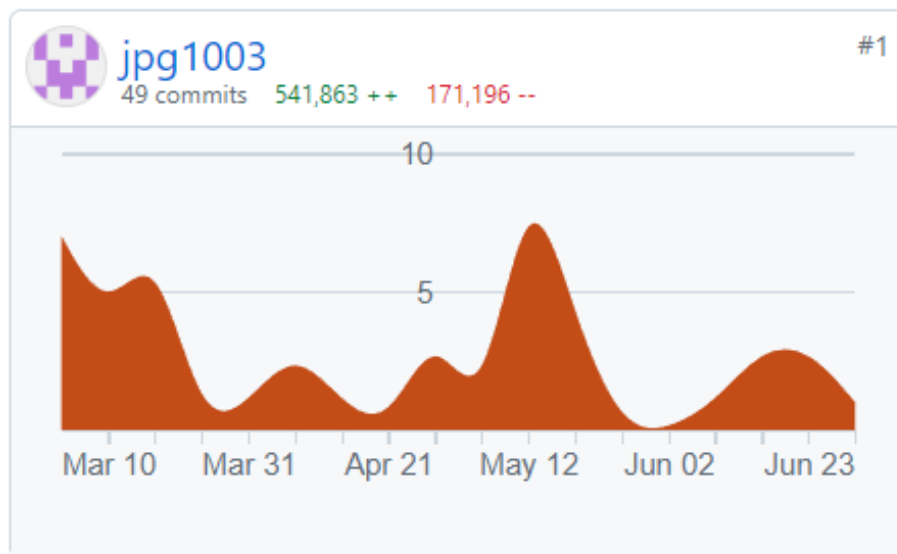


Figura A.1: Commits realizados durante la realización del TFG

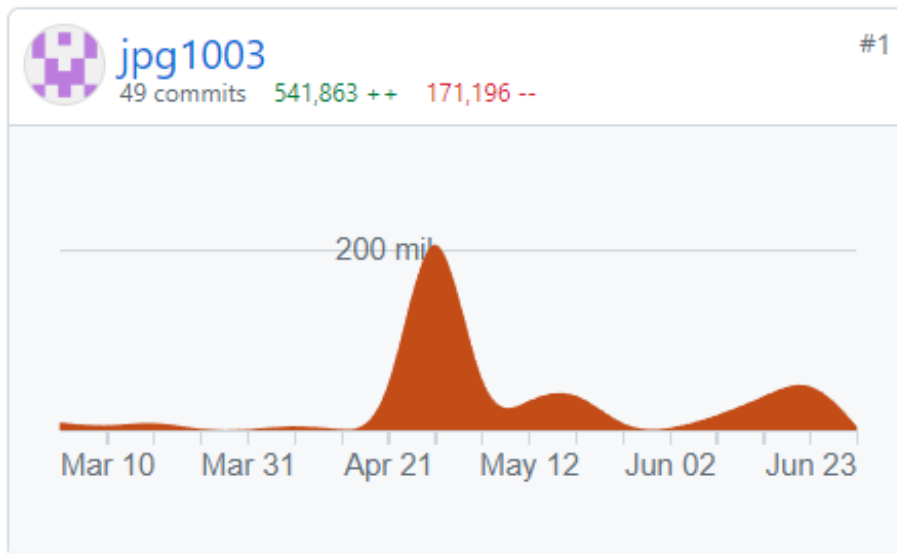


Figura A.2: Additions realizados durante la realización del TFG



Figura A.3: Deletions realizados durante la realización del TFG

Se pueden dividir en 3 grandes etapas:

- Etapa de estudio: (Febrero a Marzo)
- Etapa de desarrollo: (Abril a Mayo)
- Etapa de desarrollo y optimización: (Junio a Julio)

Etapa de estudio: (Febrero a Marzo)

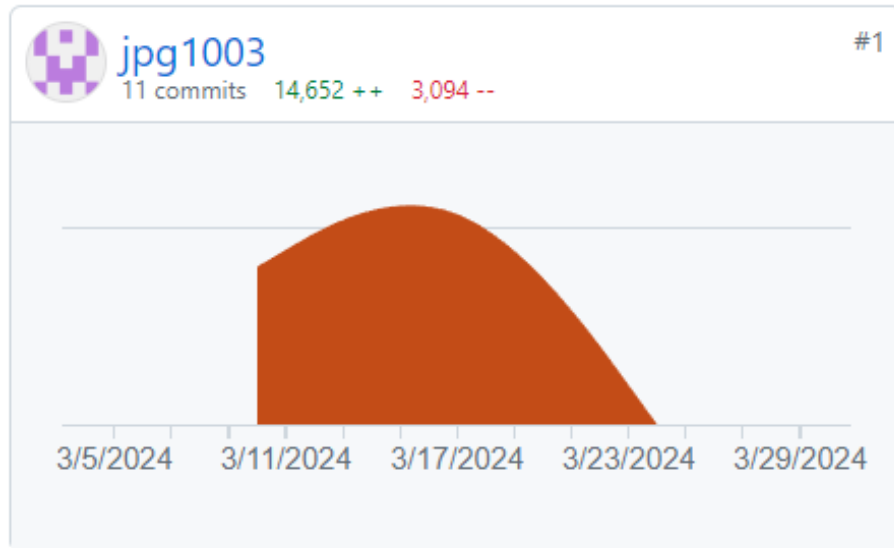


Figura A.4: Commits-Etapa 1

- **Realización de cursos online** propuestos por los tutores Bruno Baruque y Jesús Enrique Sierra. [Tutorial de Inicio de Pandas](#), [Visualización de Datos](#), [Trabajo con series temporales](#)

- **Análisis del conjunto de datos**, en archivo csv, proporcionado por los tutores. - **Creación esqueleto para la estructura el TFG en Github**. - **Definición y creación de los primeros notebooks**, principalmente análisis del conjunto de datos. - **Definición y creación de los primeros notebooks de machine learning**, al final de la etapa. - **Comentar código e imprimir comentarios en los notebooks**.

Los principales problemas o obstáculos que me encontré fueron principalmente los siguientes:

- **Tiempo invertido en la realización de los cursos**. - **Estructuración del código**. Me tomo mucho tiempo poder llegar a definir como quería mostrar el código, me decidí por un notebook principal que realizara llamadas al resto de notebooks con códigos mas específicos para cada modelo o experimento a realizar. - **Plotteos y definiciones básicas como normalizar o escalar el conjunto de datos**.

Etapa de desarrollo: (Abril a Mayo)

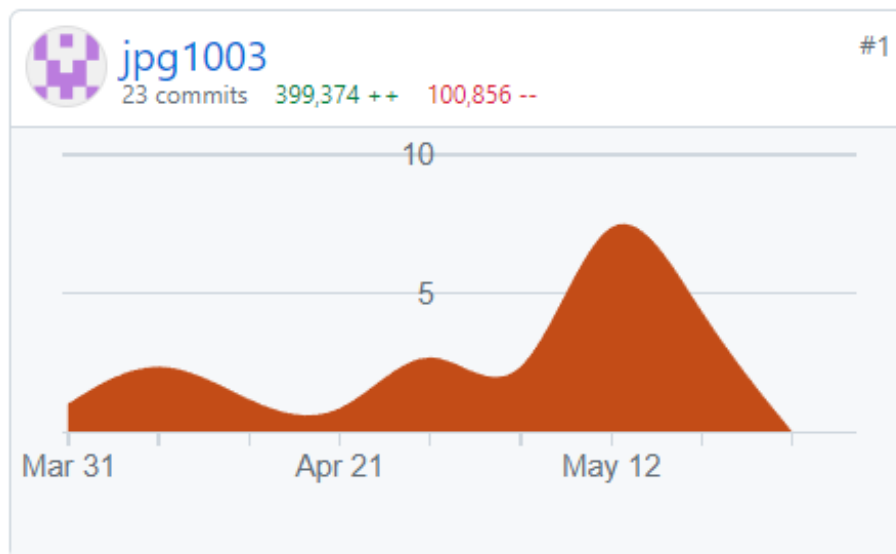


Figura A.5: Commits-Etapa 2

- Cambio de análisis del conjunto de datos, preprocessing.
- Definición y creación notebooks de machine learning.
- Definición y creación notebooks de deep learning.

Los problemas que me encontré en esta etapa fueron:

- **Compilación modelos depp learning.** Al ejecutar los primeros modelos de deep learning los datos normalizados me proporcionaban errores de compilación con modelos SRNN o LSTM, cambiando a datos escalados y shapeando los modelos pudieron ejecutarse.

- **Utilización de ventanas temporales en los modelos deep learning.** En esta etapa no supe identificar este requerimiento por parte de los tutores y estuve implementando varias formas de poder utilizar ventanas temporales en el código.

- Utilización de modelos deep learning y callbacks.

- Gráficas y definiciones básicas como normalizar o escalar el conjunto de datos.

Etapas de desarrollo y optimización: (Junio a Julio)

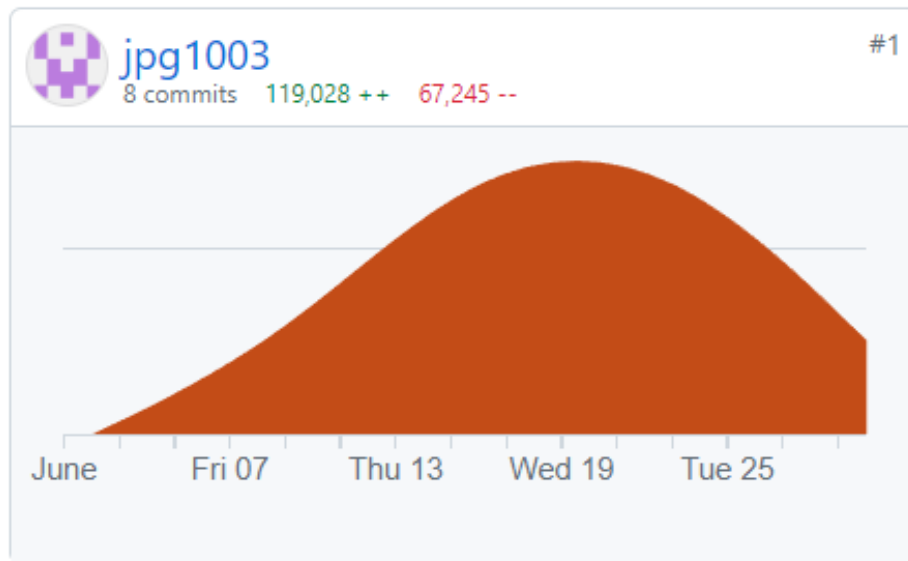


Figura A.6: Commits-Etapa 3

- **Añado nuevos gráficos** en análisis del conjunto de datos, preprocesing.
- **Definición y creación ventanas temporales acordadas con Bruno Baruque.**
- **Definición y creación nuevos notebooks de deep learning**
- **Definición y creación nuevos datos sintéticos a través de el aplicativo smote.**
- **Continuar con el comentado del código e imprimir comentarios en los notebooks.**

En la última etapa los problemas que se han acontecido son:

- **Utilización de ventanas temporales en los modelos deep learning.** Después de varias algunas reuniones con Bruno Baruque se llego a la defnicon correcta para las ventanas temporal en el conjunto de datos.
- **Utilización de modelos deep learning y callbacks.**
- **Utilización datos sintéticos.** Definir correctamente esta generacion de datos y poder utilizarlos en los modelos deep learning correctamente.

A.3. Estudio de viabilidad

Viabilidad económica

Viabilidad legal

Apéndice B

Especificación de Requisitos

B.1. Introducción

La Especificación de Requisitos tiene como objetivo definir de manera clara y detallada las necesidades y expectativas del proyecto que consta de un notebook Jupyter definido en Python, el cual invoca a otros notebooks Jupyter secundarios para realizar diversas tareas.

Este documento sirve para asegurar que todas las partes interesadas comprendan y acuerden los objetivos y funcionalidades del sistema a desarrollar.

En un entorno de trabajo basado en notebooks Jupyter, es esencial contar con una especificación precisa que guíe el desarrollo e implementación. Esto facilita la colaboración y la comunicación entre los desarrolladores y usuarios finales.

B.2. Objetivos generales

Los objetivos principales de este TFG son los siguientes:

- **Facilitar el procesamiento de datos EEG.** Generados en la universidad mediante el interfaz BCI.
- **Automatizar tareas repetitivas.** Con la creación de notebooks secundarios para poder ejecutar las tareas comunes y repetitivas relacionadas con los datos EEG.
- **Mejorar la tasa de acierto en los análisis de modelos** implementando modelos precisos y fiables.

- **Asegurar la accesibilidad y usabilidad en los notebooks.** Los notebooks son herramientas intuitivas y accesibles que aportan esa facilidad a la hora de interactuar.
- **Integración y extensibilidad.** Al utilizar llamadas a otros notebooks se asegura que se permita la integración de nuevos notebooks y se asegura posibles nuevas mejoras en el código.

B.3. Catálogo de requisitos

Hay dos tipos de requisitos, los funcionales (qué debe hacer el código) y los no funcionales (cómo debe funcionar el código):

Requisitos funcionales

- **RF-001 Iniciar notebook Main:**
 - **Descripción:** Los usuarios han de poder iniciar el notebook principal para poder comenzar con la ejecución del trabajo.
 - **Prioridad:** Alta
 - **Criterios de aceptación:** El notebook Main debe abrirse correctamente en menos de 5 segundos.
- **RF-002 Invocación de Notebooks Secundarios Preparatorios:**
 - **Descripción:** Los usuarios han de poder invocar los notebook secundarios preparatorios para poder ejecutar con seguridad el resto de notebooks de preprocesado y modelados.
 - **Prioridad:** Alta
 - **Criterios de aceptación:** Los notebooks secundarios preparatorios se han de ejecutar correctamente y que la celda en la que se llama al notebook secundario no devuelva errores. Los notebooks preparatorios. Su ejecución no ha de ser mayor a 5 minutos.
- **RF-003 Invocación de Notebook Secundarios carga de archivo CSV:**
 - **Descripción:** Los usuarios han de poder invocar los notebook secundarios carga de archivo CSV, para la ejecución de este notebook secundario se ha de subir un archivo csv con el conjunto de datos a analizar y una vez subido el archivo, se iluminara un botón con nombre 'Procesar datos' que procesara los datos una vez pulsado.

- **Prioridad:** Alta
 - **Criterios de aceptación:** Si no se ejecuta correctamente la llamada a este notebook no se podrá seguir con la ejecución del resto de notebooks puesto que sin datos no se podrían realizar las siguientes acciones.
- **RF-004 Invocación de Notebooks Secundarios análisis y pre-processing:**
 - **Descripción:** Los usuarios han de poder invocar los notebook secundarios análisis y preprocessing para poder proporcionar este procesado de datos antes de la ejecución de modelos machine o deep learning.
 - **Prioridad:** Alta
 - **Criterios de aceptación:** Si no se ejecuta correctamente la llamada a este notebook no se podrá seguir con la ejecución del resto de notebooks puesto que sin la realización del preprocessing de los datos no se generan los csv correspondientes para poder seguir ejecutando los siguientes notebooks.
 - **RF-005 Invocación de Notebooks Secundarios para experimentos:**
 - **Descripción:** Los usuarios han de poder invocar los notebook secundarios para experimentos y poder generar los datos sobre Tasa de acierto y generación de matrices de confusión.
 - **Prioridad:** Media
 - **Criterios de aceptación:** Estos notebooks se puede ejecutar de manera individual o en el orden pre establecido. para realizar el análisis final lo ideal es ejecutar todos los notebooks de esta session.
 - **RF-006 Invocación de Notebook Secundario resultados:**
 - **Descripción:** Los usuarios han de poder invocar los notebook Secundario resultados y poder visualizar los resultados de las particiones test del conjunto de datos. Se han de haber ejecutado todos los notebooks de esta sección.
 - **Prioridad:** Alta
 - **Criterios de aceptación:** Sin la ejecución de este notebook no se obtendrían los resultados del trabajo y por lo tanto es crítico este último notebook para poder visualizar la tasa de acierto en datos que el modelo no conocía en la etapa de entrenamiento y validación.

Requisitos no funcionales

- **RNF-001 Rendimiento:**
 - **Descripción:** El sistema donde se ejecuten los notebooks ha de poder manejar la ejecución sin causar demoras significativas.
 - **Criterios de aceptación:** El tiempo de ejecución para la ejecución de cada notebook secundario no ha de ser superior a 15 minutos.
- **RNF-002 Usabilidad:**
 - **Descripción:** Los notebooks han de ser fáciles de entender y usar por los usuarios.
 - **Criterios de aceptación:** La interfaz de los notebooks ha de ser intuitiva y clara.
- **RNF-003 Escalabilidad:**
 - **Descripción:** En la configuración del notebooks principal se debe permitir la adhesión de nuevos notebooks y sus llamadas desde el notebook principal.
 - **Criterios de aceptación:** Poder seguir integrando nuevos notebooks para ser llamados desde el notebook principal sin afectar al funcionamiento del código.

Restricciones

- **R-001:** El sistema debe operar en un entorno Jupyter Notebook.
- **R-002:** Todos los notebooks secundarios deben estar disponibles en el mismo entorno de ejecución que el notebook principal.
- **R-003:** El procesamiento y análisis de datos debe realizarse utilizando bibliotecas compatibles con Python 3.x.

B.4. Especificación de requisitos

Voy a describir cada caso de uso identificado:

CU-001	Iniciar notebook principal (Main)
Versión	1.0
Autor	José Luis Pérez Gómez
Requisitos asociados	RF-001
Descripción	El notebook principal necesita ser ejecutado para iniciar el análisis del conjunto de datos.
Precondición	R-001, R-002, R-003
Acciones	<ol style="list-style-type: none"> 1. Acceder a la carpeta donde se encuentren los notebooks. 2. Seleccionar el notebook '1.Main.ipynb' y hacer doble clic sobre el archivo.
Postcondición	El usuario podrá visualizar el código del notebook principal
Excepciones	
Importancia	Alta

Tabla B.1: CU-001 Iniciar notebook principal (Main).

CU-002	Ejecución celda 1. Crear un entorno virtual.
Versión	1.0
Autor	José Luis Pérez Gómez
Requisitos asociados	RF-002
Descripción	Desde el notebook principal se ejecuta la primera celda que llama a un notebook secundario.
Precondición	R-001, R-002, R-003 y Haber iniciado correctamente el notebook principal (Main).
Acciones	Pulsar Run en la celda ejecutable desde el notebook principal (1. Crear un entorno virtual)
Postcondición	La salida de la ejecución no de ningún error
Excepciones	
Importancia	Media

Tabla B.2: CU-002 Ejecución celda 1. Crear un entorno virtual.

CU-003	Ejecución celdas del notebook Main (Notebooks Preparatorios).
Versión	1.0
Autor	José Luis Pérez Gómez
Requisitos asociados	RF-002
Descripción	Desde el notebook principal se ejecutan las tres siguientes celdas ejecutables desde el notebook principal Main.
Precondición	R-001, R-002, R-003 y Haber iniciado correctamente el notebook principal (Main)
Acciones	<ol style="list-style-type: none"> 1. Pulsar Run en la celda ejecutable "2. Instalación de bibliotecas." 2. Pulsar Run en la celda ejecutable "3. Importación de las bibliotecas instaladas." 3. Pulsar Run en la celda ejecutable "4. Asignación de valores a variables y creación de funciones."
Postcondición	La salida de la ejecución en las tres celdas no debe tener ningún error
Excepciones	
Importancia	Alta

Tabla B.3: CU-003 Ejecución celdas (Notebooks Preparatorios).

CU-004	Ejecución celda del notebook Main carga de archivo CSV.
Versión	1.0
Autor	José Luis Pérez Gómez
Requisitos asociados	RF-003
Descripción	Desde el notebook principal se ejecutan las tres siguientes celdas ejecutables desde el notebook principal Main.
Precondición	R-001, R-002, R-003, haber iniciado correctamente el notebook principal (Main) y haber ejecutado todas las celdas anteriores a esta celda
Acciones	<ol style="list-style-type: none"> 1. Pulsar Run en la celda ejecutable notebook carga de archivo CSV con el conjunto de datos a analizar: 2. Pulsar sobre el botón nuevo llamado "Upload". 3. Seleccionar el archivo de datos a analizar (para esta ejecución datosEEGTotal.csv) 4. Pulsar sobre el botón nuevo llamado "Procesar Datos".
Postcondición	La salida de la ejecución ha de imprimir por pantalla los datos a analizar.
Excepciones	
Importancia	Alta

Tabla B.4: CU-004 Ejecución celdas cargar de archivo CSV.

CU-005	Ejecución celdas del notebook Main (Notebooks análisis y preprocessing)
Versión	1.0
Autor	José Luis Pérez Gómez
Requisitos asociados	RF-004
Descripción	Desde el notebook principal se ejecutan las dos siguientes celdas ejecutables desde el notebook principal Main debajo de Notebooks análisis y preprocessing de conjunto de datos
Precondición	R-001, R-002, R-003, haber iniciado correctamente el notebook principal (Main) y haber ejecutado todas las celdas anteriores a esta celda
Acciones	<ol style="list-style-type: none"> 1. Pulsar Run en la celda ejecutable notebook carga de archivo CSV con el conjunto de datos a analizar 2. Pulsar Run en la celda ejecutable "5. Análisis de datos inicial." 3. Pulsar Run en la celda ejecutable "6. Preprocessing del conjunto de datos."
Postcondición	La salida de las ejecuciones ha de imprimir por pantalla los datos analizados y preprocesados del conjunto de datos sin errores.
Excepciones	
Importancia	Alta

Tabla B.5: CU-005 Ejecución celdas (Notebooks análisis y preprocessing)

CU-006	Ejecución celdas del notebook Main (Notebooks experimentos)
Versión	1.0
Autor	José Luis Pérez Gómez
Requisitos asociados	RF-005
Descripción	Desde el notebook principal se ejecutan las dos siguientes celdas ejecutables desde el notebook principal Main debajo de Notebooks análisis y preprocessing de conjunto de datos
Precondición	R-001, R-002, R-003, haber iniciado correctamente el notebook principal (Main) y haber ejecutado todas las celdas anteriores a esta celda
Acciones	<ol style="list-style-type: none"> 1. Pulsar Run en la celda ejecutable 7. Implementación modelos Machine Learning 2. Pulsar Run en la celda ejecutable 8.1 MLP 3. Pulsar Run en la celda ejecutable 8.2 SRNN 4. Pulsar Run en la celda ejecutable 8.3 LSTM 5. Pulsar Run en la celda ejecutable 8.4 SRNN Sliding Windows 6. Pulsar Run en la celda ejecutable 8.5 LSTM Sliding Windows 7. Pulsar Run en la celda ejecutable 9.1 Aumento de datos con SMOTE 8. Pulsar Run en la celda ejecutable 9.2 SRNN Sliding Windows 9. Pulsar Run en la celda ejecutable 9.3 LSTM Aumento de datos 10. Pulsar Run en la celda ejecutable 9.4 SRNN Aumento de datos Sliding Windows 11. Pulsar Run en la celda ejecutable 9.5 LSTM Aumento de datos Sliding Windows
Postcondición	La salida de las ejecuciones ha de imprimir por pantalla los datos Tasa de acierto y matrices de confusión.
Excepciones	
Importancia	Alta

Tabla B.6: CU-006 Ejecución celdas (Notebooks experimentos)

CU-007	Ejecución celda del notebook Main (Notebook recopilado de resultados)
Versión	1.0
Autor	José Luis Pérez Gómez
Requisitos asociados	RF-006
Descripción	Desde el notebook principal se ejecutan las dos siguientes celdas ejecutables desde el notebook principal Main debajo de Notebooks análisis y preprocessing de conjunto de datos
Precondición	R-001, R-002, R-003, haber iniciado correctamente el notebook principal (Main) y haber ejecutado todas las celdas anteriores a esta celda
Acciones	<ol style="list-style-type: none"> 1. Pulsar Run en la celda ejecutable Notebook recopilado de resultados
Postcondición	La salida de las ejecuciones ha de imprimir por pantalla los datos Tasa de acierto para el conjunto de datos validación y test en todos los experimentos.
Excepciones	
Importancia	Alta

Tabla B.7: CU-007 Ejecución celdas (Notebook resultados)

Apéndice C

Especificación de diseño

C.1. Introducción

La especificación de datos es un componente crítico en el desarrollo de sistemas de información, especialmente cuando se trabaja con conjuntos de datos complejos como los datos EEG.

Mediante BCI, sistema que permite mediante adquisición de señales EEG, se han podido interpretar las señales adquiridas a través de las señales EEG y poder transformarlas en un conjunto de datos para su posterior análisis.

Descripción de los Datos:

El archivo datosEEGTotal.csv contiene los datos facilitados para poder realizar los experimentos para la ejecución del TFG.

Su formato es de tipo CSV con un separador (;) entre los datos que lo componen.

Las características recogidas en el archivo de los datos EEG son las siguientes:

Timestamp, Attention, Meditation, Delta, Theta, LowAlpha, HighAlpha, LowBeta, HighBeta, LowGamma, HighGamma, Signal y Key.

Timestamp: Registro de tiempo para los experimentos, medido en mili-segundos.

Attention: Registra el grado de atención del participante que realiza el experimento.

Meditation: Grado de calma que tendría el individuo.

Delta: Son ondas de baja frecuencia (1 y 4 Hz), están presentes en etapas de sueño profundo, durante una meditación profunda y en pacientes con lesiones cerebrales o con TDAH severo.

Theta: Ondas entre 4 y 8 Hz, se encuentran en estados de calma profunda y sueño R.E.M., están ligadas al aprendizaje, memoria y intuición.

Alpha: Ondas entre 8 y 12 Hz, representan un estado de poca actividad cerebral y se asocian a un estado de calma mental. Divididas en dos señales LowAlpha y HighAlpha

Beta: Se diferencian en LowBeta y HighBeta, su frecuencia esta entre 12 y 35Hz, asociadas a una alta actividad mental.

Gamma: En los datos se diferencia LowGamma y HighGamma, son ondas por encima de 30Hz y suelen aparecer cuando hay una alta concentración o atención

Signal: Podría ser la señal de que aporta la interfaz BCI.

Key: Valores target de lo que el individuo estaba pensando o visualizando durante el experimento.

La transformación de datos o el preprocessing que se ha realizado para poder afrontar los experimentos del TFG han sido los siguientes:

Unificación de características Key: El conjunto de datos tiene varios valores en Key que indican los mismo. LButton y Left.

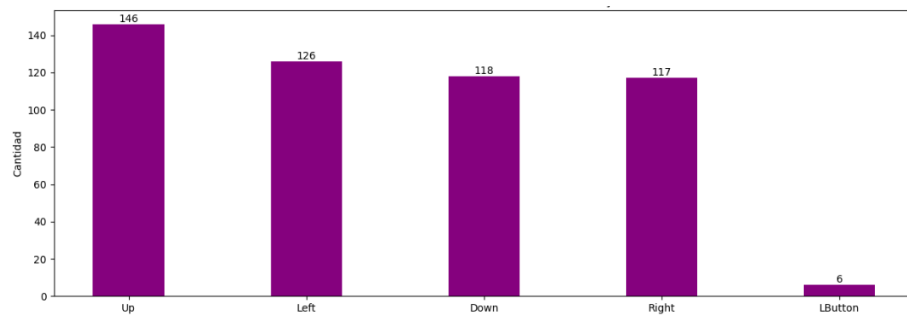


Figura C.1: Cantidad de elementos para la característica Key

División del conjunto de datos: El conjunto de datos tiene cuatro segmentos divididos por su Timestamp, superponiéndose entre ellos. Divido en estos cuatro segmentos para poder realizar experimentos y también de

el conjunto de datos sin dividir para realizar experimentos conjuntos a los cuatro segmentos.

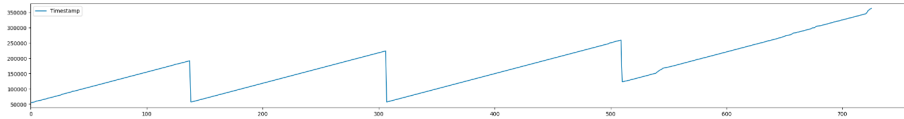


Figura C.2: Análisis característica Timestamp del conjunto de datos

Eliminación de características: Elimino las características Signal por no aportar nada significativo en el conjunto de datos y Timestamp porque no quiero que los datos aportados puedan tener una patrón temporal que haga que los experimentos no sean reales.

Eliminación de outliers: Elimino los outliers (datos atípicos) del conjunto de datos. Utilizo zcore con un umbral de 3. **Regla del 68-95-99.7**

Escalado de datos: He utilizado la opción de escalar los datos ya que no son datos normales puesto que no tienen una distribución gaussiana en sus datos.

C.2. Diseño de datos

En este trabajo con datos EEG, el diseño de datos consta de:

Diagramas entidad-relación (ERD)

Relaciones y Dependencias:

C.3. Diseño procedimental

En este trabajo he utilizado notebooks Jupyter para analizar el conjunto de datos EEG, por este motivo el diseño arquitectónico resultante ha sido el siguiente:

Arquitectura del Sistema:

La idea principal era que la ejecución de varios modelos de machine y deep learning se pudieran ejecutar de una manera eficiente y que fueran independientes sus ejecuciones. Para esto, el diseño se compone de un notebook jupyter que llama celda a celda a diferentes notebooks en los cuales se desarrolla el código a ejecutar.

Los componentes principales son:

- **Notebook principal**, desde el cual se orquesta todas las ejecuciones de los demás notebooks.
- **Notebooks secundarios preparatorios**, que como su nombre indica, preparan el entorno para que puedan ser ejecutados los experimentos.
- **Notebook subida datos en archivo CSV**, desde esta llamada se sube el archivo con el conjunto de datos a analizar.
- **Notebook secundarios para experimentos** donde se ejecutan los experimentos asociados al TFG.
- **Notebook de resultados** que imprime los resultados de tasa de acierto en los experimentos contra los datos de tipos test.

Las tecnologías y Herramientas utilizadas han sido las siguientes:

Manejo de datos y preprocessing:

- **Pandas**: Para la manipulación y análisis de datos estructurados en DataFrames.
- **NumPy**: Para operaciones numéricas eficientes en matrices, arrays y listas.
- **Scikit-learn y SciPy**: Para escalado de datos, transformación y técnicas de preprocesamiento como StandardScaler y z-score.

Técnicas de validación:

- **Holdout**: División de los datos en conjuntos de entrenamiento, validación y prueba para evaluación inicial.
- **K-Fold Cross-Validation**: Validación cruzada con KFold para evaluar la variabilidad del modelo.
- **Leave-One-Out**: Validación de uno en uno los datos dejando los demás fuera, así se evalúa la robustez del modelo.

Modelos de aprendizaje automático:

- **K-Nearest Neighbors (KNN)**: Clasificación basada en la proximidad de los puntos de los datos.
- **Árboles de decisión (Decision Tree)**: Utiliza una estructura de árbol para realizar predicciones basadas en decisiones binarias.
- **Random Forest**: Mejora la precisión y reduce el sobreajuste utilizando múltiples árboles de decisión.

Redes Neuronales y Modelos Avanzados:

- **Multi-Layer Perceptron (MLP):** Implementan redes neuronales feedforward. Esta arquitectura de red neuronal no tiene ciclos en sus conexiones neuronales sino que los datos van en una dirección desde la capa de entrada a la capa de salida. Es el más básico.
- **Simple Recurrent Neural Network (SRNN):** Es un tipo básico de red neuronal que tiene conexiones recurrentes que le permite mantener y utilizar información previa en la secuencia de datos. Es muy parecida a MLP pero en este caso las conexiones entre neuronas si pueden ir hacia atrás.
- **Long Short-Term Memory (LSTM):** Es una red neuronal avanzada en comparación con la recurrente SRNN. La diferencia principal es que puede retener la información relevante durante mas tiempo que SRNN.

Técnicas para la creación de datos sintéticos:

- **SMOTE (Synthetic Minority Over-sampling Technique):** Uso de la biblioteca imblearn para crear ejemplos sintéticos.

Infraestructura utilizada?

C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

D.1. Introducción

Este TFG de análisis de un conjunto de datos EEG se centra en la aplicación de técnicas avanzadas de aprendizaje automático utilizando notebooks Python. El objetivo principal es procesar y analizar datos EEG para identificar patrones y correlaciones significativas que puedan revelar comportamientos específicos en la actividad cerebral a la hora de realizar un experimento el individuo.

El propósito fundamental es aplicar modelos de aprendizaje automático para mejorar la comprensión de las señales EEG, los datos aportados por la universidad de burgos constan de unos experimentos realizados con BCI y varios voluntarios es los que estos usuarios deben visualizar una serie de imagenes compuestas por direcciones, arriba, abajo, derecha e izquierda.

Este trabajo podría tener aplicaciones importantes en tecnologías de interfaz cerebro-computadora (BCI) aplicadas a por ejemplo, control de sillas de ruedas, control de dispositivos electrónicos, realidad virtual, videojuegos, control de robots, etc...

D.2. Estructura de directorios

La estructura de los directorios del trabajo es la siguiente:

- **/documentacion/**: Esta carpeta contiene toda la documentación relacionada con el TFG.
- **/documentacion/imagenes/**: Archivo de imágenes para todo el proyecto, documentación y código.
- **/documentacion/latex**: Documentación en formato latex y PDF.
- **/codigo**: Carpeta para archivar notebooks y datos para la ejecución del código.
- **/codigo/datos/**: Archivo datosEEGTotal.csv con el origen de datos.
- **/codigo/datos/csv/**: Archivos de datos csv auto generados tras las ejecuciones de algunos notebooks.
- **/codigo/notebooks**: Contiene los notebooks Jupyter y por lo tanto, el código.

D.3. Manual del programador

Este manual sera utilizado principalmente por las personas involucradas en futuros cambios, mejoras o nuevos desarrollos en el código.

Para ello, detallo las funciones de cada uno de los componentes :

D.4. Compilación, instalación y ejecución del proyecto

D.5. Pruebas del sistema

Apéndice E

Documentación de usuario

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario

Apéndice F

Anexo de sostenibilización curricular

F.1. Introducción

Este anexo incluirá una reflexión personal del alumnado sobre los aspectos de la sostenibilidad que se abordan en el trabajo. Se pueden incluir tantas subsecciones como sean necesarias con la intención de explicar las competencias de sostenibilidad adquiridas durante el alumnado y aplicadas al Trabajo de Fin de Grado.

Más información en el documento de la CRUE https://www.crue.org/wp-content/uploads/2020/02/Directrices_Sostenibilidad_Crue2012.pdf.

Este anexo tendrá una extensión comprendida entre 600 y 800 palabras.