

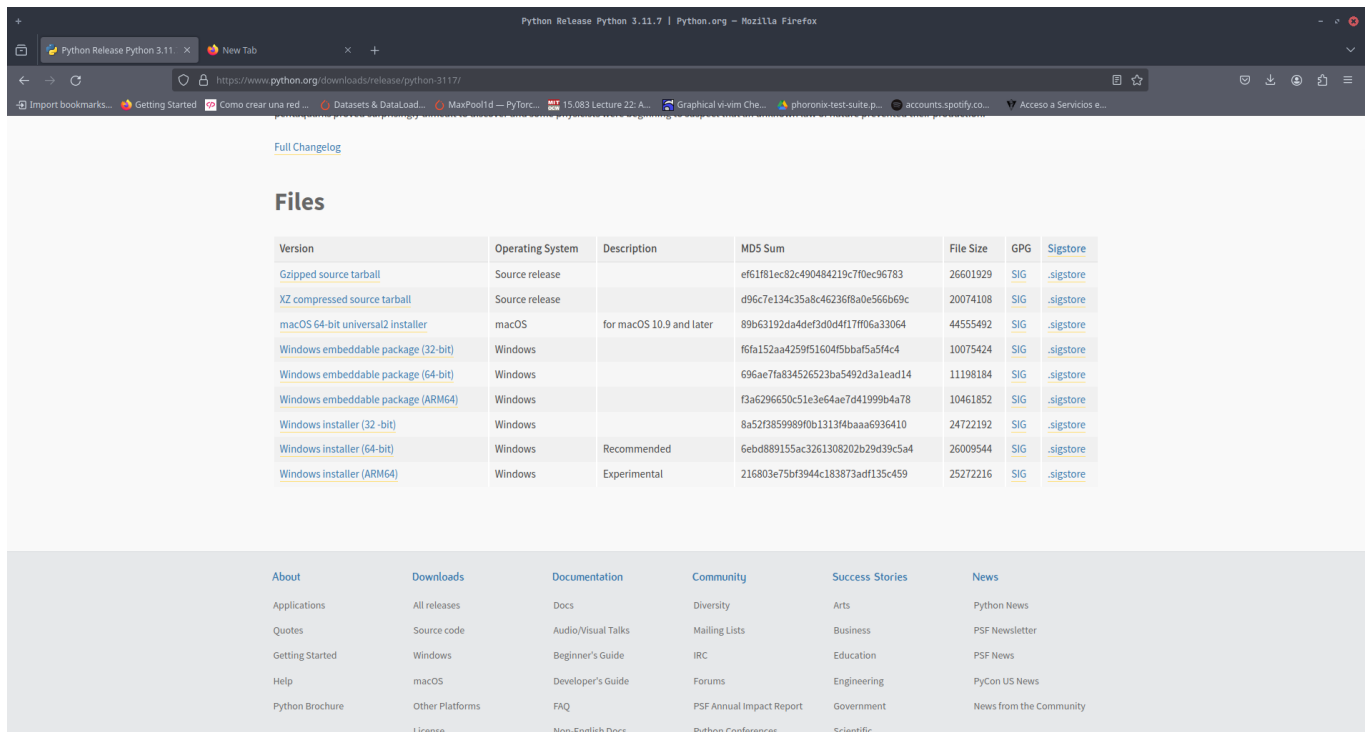
README

Programa para Servicio social

Requerimientos

Instalación de python 3.11.4

Se puede descargar el instalador en <https://www.python.org/downloads/release/python-3117/>



Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore
Gzipped source tarball	Source release		ef61f81ec82c490484219c7f0ec96783	26601929	SIG	.sigstore
XZ compressed source tarball	Source release		d96c7e134c35a8c46236f8a0e566b69c	20074108	SIG	.sigstore
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	89b63193da4def3d0d4f17f06a33064	44555492	SIG	.sigstore
Windows embeddable package (32-bit)	Windows		f6fa152aa4259f51604f5bbaf5a5f4c4	10075424	SIG	.sigstore
Windows embeddable package (64-bit)	Windows		696ae7fa834526523ba5492d3a1ead14	11198184	SIG	.sigstore
Windows embeddable package (ARM64)	Windows		f3a6296650c51e3e64ae7d41999b4a78	10461852	SIG	.sigstore
Windows installer (32-bit)	Windows		8a52f3859989f0b13134baa6936410	24722192	SIG	.sigstore
Windows installer (64-bit)	Windows	Recommended	6ebd889155ac3261308202b29d39c5a4	26009544	SIG	.sigstore
Windows installer (ARM64)	Windows	Experimental	216803e75bf3944c183873ad135c459	25272216	SIG	.sigstore

About

Downloads

Documentation

Community

Success Stories

News

Applications

Quotes

Getting Started

Help

Python Brochure

All releases

Source code

Windows

macOS

Other Platforms

License

Docs

Audio/Visual Talks

Beginner's Guide

Developer's Guide

FAQ

Non-English Docs

Diversity

Mailing Lists

IRC

Forums

PSF Annual Impact Report

Python Conferences

Arts

Business

Education

Engineering

Government

Scientific

Python News

PSF Newsletter

PSF News

PyCon US News

News from the Community

Elegir el recomendado: Windows installer (64-bit)

Ejecutar ambiente

Para ejecutar cualquiera de los programas es necesario iniciar un ambiente virtual de python

Crear entorno virtual:

```
python -m venv tutorial-env
```

Una vez creado el entorno virtual, podrás activarlo.

En Windows, ejecuta:

```
tutorial-env/Scripts/activate
```

En Unix o MacOS, ejecuta:

```
source tutorial-env/bin/activate
```

Para más detalles consultar en <https://docs.python.org/es/3/tutorial/venv.html>

Luego instalar requerimientos en Aplicacion\requi.txt

Una vez activo el ambiente, instalar requerimientos:

```
pip install -r Aplicacion/requi.txt
```

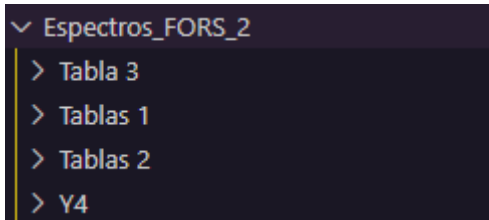
Utilizando script de python

Ejecutar desde la carpeta del proyecto:

```
python Aplicacion/setup.py
```

Actualizar Base de Datos para el programa

Para ingresar una nueva tabla de mediciones es necesario incluir la nueva carpeta dentro de Espectros_FORS_2. En la misma altura que Tabla 3, Tablas 1, etc.



De forma que la estructura es la siguiente

```
Espectros_FORS_2
├── Tabla 3
├── Nueva_carpeta
│   ├── n1espectro.asd / n1espectro.asd.txt
│   ├── n2espectro.asd / n2espectro.asd.txt
│   └── ...
```

Y la ruta a un espectro será: Espectros_FORS_2/Nueva_carpeta/n1espectro.asd.txt

Crear un archivo csv con la siguiente estructura:

|

Carpeta,Tabla,Espectro,Pigmento,Aglutinante,Base de preparaci
Tablas1,C1,A100000.asd,Ocre de mina ingles,Aceite de linaza,Carbonato de calcio

Ejecutar el archivo actualizaCSV_DB.py, e introducir el nombre del archivo csv que debe estar en la misma carpeta que actualizaCSV_DB.py

```
(ss) >>> Aplicacion git:(entrega) x python actualizaCSV_DB.py
Recuperando espectros
Introducir nombre del archivo csv:
nombre.csv
```

PROGRAMA:ORGANIZADOR

El objetivo de este programa es proveer con una interfaz para mantener organizadas mediciones de espectros FORS proporcionando filtros y la posibilidad de gráficar y combinar gráficas. Notemos un ejemplo para utilizar los filtros:

Filtro	Descripción	Ejemplo
Carpeta	El nombre de la carpeta donde se encuentran las tablas	Tabla 3
Tabla	El nombre, o etiqueta, de la tabla	C1
Espectro	Nombre del archivo	A100000.asd
Pigmento	Pigmento del espectro	Ocre de mina ingles
Aglutinante	Aglutinante del espectro	Aceite de linaza
Base_de_preparacion	Base de preparación del espectro	Carbonato de calcio

PROGRAMA:CLASIFICADOR

El objetivo de este programa es proveer con una interfaz que permita subir un archivo de una medición FORS en formato asd o txt y realizar una clasificación a partir del entranamiento realizado sobre la base de datos del organizador.

El programa provee de 6 algoritmos de clasificación:

Regresion logistica

En estadística, el modelo logístico (o modelo logit) se utiliza para modelar la probabilidad de que exista una determinada clase o evento como pasa / no pasa, gana / pierde, vivo / muerto o sano / enfermo. (Tiene dado pigmento, dado aglutinante)

La regresión logística analiza datos distribuidos binomialmente de la forma

$$Y_i \sim B(p_i, n_i), \text{ para } i = 1, \dots, m,$$

donde los números de ensayos Bernoulli n_i son conocidos y las probabilidades de éxito p_i son desconocidas.

El modelo es entonces obtenido a base de lo que cada ensayo (valor de i) y el conjunto de variables explicativas/independientes puedan informar acerca de la probabilidad final. Estas variables explicativas pueden pensarse como un vector

X_i k -dimensional y el modelo toma entonces la forma:

$$p_i = E\left(\frac{Y_i}{n_i} \middle| X_i\right).$$

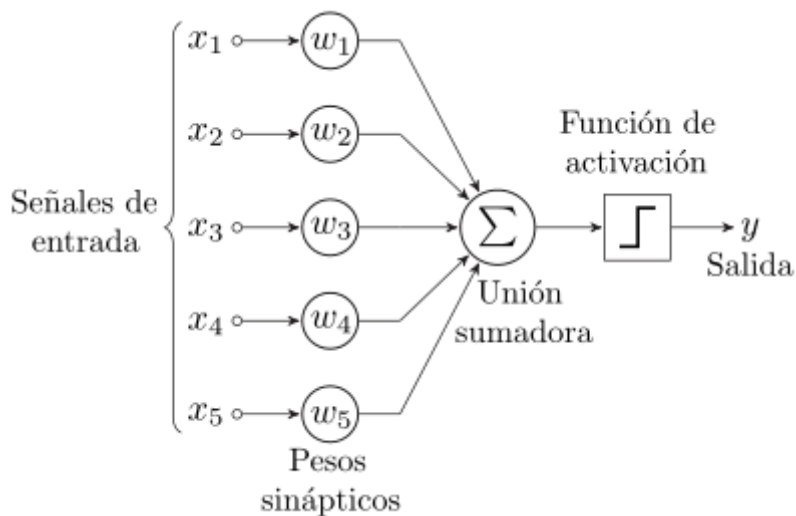
Los logits de las probabilidades binomiales desconocidas (i.e., los logaritmos de la razón de momios) son modeladas como una función lineal de los X_i .

$$\text{logit}(p_i) = \ln\left(\frac{p_i}{1 - p_i}\right) = \beta_0 + \beta_1 x_{1,i} + \dots + \beta_k x_{k,i}.$$

Note que un elemento particular de X_i puede ser ajustado a 1 para todo i obteniéndose una constante independiente en el modelo. Los parámetros desconocidos β_j son usualmente estimados a través del método de máxima verosimilitud.

Perceptron

En el campo de las Redes Neuronales, el perceptrón, creado por Frank Rosenblatt, se refiere a: la neurona artificial o unidad básica de inferencia en forma de discriminador lineal, a partir de lo cual se desarrolla un algoritmo capaz de generar un criterio para seleccionar un subgrupo a partir de un grupo de componentes más grande.



Regresión lineal

En estadística, la regresión lineal o ajuste lineal es un modelo matemático usado para aproximar la relación de dependencia entre una variable dependiente Y , m variables independientes

X_i con

$m \in \mathbb{Z}^+$ y un término aleatorio ε .

Este modelo puede ser expresado como:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_m X_m + \varepsilon$$

donde:

Y es la variable dependiente o variable de respuesta.

X_1, X_2, \dots, X_m son las variables explicativas, independientes o regresoras.

$\beta_0, \beta_1, \beta_2, \dots, \beta_m$ son los parámetros del modelo, miden la influencia que las variables explicativas tienen sobre el regrediendo.

el término β_0 es la intersección o término "constante", las β_i ($i \geq 1$) son los parámetros respectivos a cada variable independiente, y m es el número de parámetros independientes a tener en cuenta en la regresión.

Naive Bayes classifier

Un clasificador de Naive Bayes asume que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de cualquier otra característica, dada la clase variable. Por ejemplo, una fruta puede ser considerada como una manzana si es roja, redonda y de alrededor de 7 cm de diámetro. Un clasificador de Naive Bayes considera que cada una de estas características contribuye de manera independiente a la probabilidad de que esta fruta sea una manzana, independientemente de la presencia o ausencia de las otras

características.

El clasificador Bayer (la función `classify`) se define como:

$$\text{classify}(f_1, \dots, f_n) = \underset{c}{\operatorname{argmax}} p(C = c) \prod_{i=1}^n p(F_i = f_i | C = c).$$

C-Support Vector Classification.

Una SVM es un modelo que representa a los puntos de muestra en el espacio, separando las clases a 2 espacios lo más amplios posibles mediante un hiperplano de separación definido como el vector entre los 2 puntos, de las 2 clases, más cercanos al que se llama vector soporte. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de los espacios a los que pertenezcan, pueden ser clasificadas a una o la otra clase.

Más formalmente, una SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación o regresión. Una buena separación entre las clases permitirá una clasificación correcta.

Random Forest

Random forest (o random forests) también conocidos en castellano como Bosques aleatorios es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. Es una modificación sustancial de bagging que construye una larga colección de árboles no correlacionados y luego los promedia.

Cada árbol es construido usando el siguiente algoritmo:

1. Sea N el número de casos de prueba, M es el número de variables en el clasificador.
2. Sea m el número de variables de entrada a ser usado para determinar la decisión en un nodo dado; m debe ser mucho menor que M
3. Elegir un conjunto de entrenamiento para este árbol y usar el resto de los casos de prueba para estimar el error.
4. Para cada nodo del árbol, elegir aleatoriamente m variables en las cuales basar la decisión. Calcular la mejor partición del conjunto de entrenamiento a partir de las m variables.

Para la predicción un nuevo caso es empujado hacia abajo por el árbol. Luego se le asigna la etiqueta del nodo terminal donde termina. Este proceso es iterado por todos los árboles en el

ensamblado, y la etiqueta que obtenga la mayor cantidad de incidencias es reportada como la predicción.