

## Guía de uso – Proyecto Clasificación de portafolios de proyectos

### Tabla de contenido

1. Requisitos:	1
2. Formato de entrada:	1
3. Formato de salida:	4
4. Guía de ejecución:	4
5. Uso de Biblioshiny	5
6. Problemas frecuentes:	6

### 1. Requisitos:

- Python 3.11 (versión recomendada: <https://www.python.org/downloads/release/python-3117/>). Durante la instalación, asegúrese de que se añade el ejecutable al PATH.

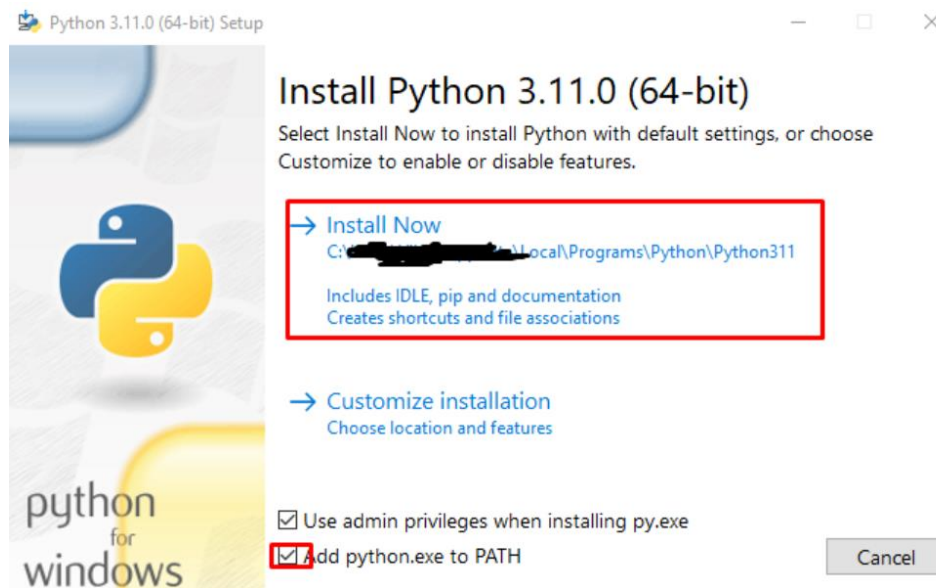


Figura 1: Pestaña inicial de instalación de Python, donde se añade el ejecutable de Python al PATH

- R (versión recomendada: <https://cran.r-project.org/bin/windows/base/>)
- R Studio (versión recomendada: <https://posit.co/download/rstudio-desktop/>)

### 2. Formato de entrada:

data/dataToClassify: Debe ser un archivo CSV (separado por comas) en formato UTF-8. Ya sea información importada desde Scopus o provista por el usuario, requiere de las siguientes columnas en cada registro:

- Title (título del artículo o proyecto)
- Abstract (resumen del artículo o proyecto)
- Author Keywords (palabras clave provistas por el autor del artículo o proyecto)
- Index Keywords (palabras clave generales del artículo o proyecto)

Los campos “Author Keywords” e “Index Keywords” pueden dejarse vacíos, pero es recomendable llenarlos para que la clasificación sea más precisa.

Por otro lado, si desea utilizar una herramienta de análisis bibliográfico para analizar los documentos importados de Scopus, seleccione todas las columnas que Scopus tiene disponibles durante la exportación y lea la sección “5. Uso de biblioshiny”.

data/verifiedArticles: Debe ser un archivo CSV (separado por comas) en formato UTF-8 Los artículos que se usan como fuente de entrenamiento poseen 4 columnas para cada registro:

- Title (título del artículo)
- Abstract (resumen del artículo)
- Category (categoría asignada al artículo)
- Criteria (criterio asignado al artículo)

Si se desea extender el número de registros clasificados, es necesario seguir la convención del marco de trabajo definida por Héctor Julio Melo Ocampo:

Category	Criteria	Question
<b>Strategic alignment</b>	Alignment with strategy and objectives	How does the use of analytics impact the alignment of portfolio management with the organization's strategy and objectives?
	Selection	What is the impact of analytics on the selection process in portfolio management?
	Prioritization	How does the use of analytics affect the prioritization of projects in portfolio management?
	Portfolio optimization/balance	To what extent does analytics contribute to portfolio optimization and balance?
<b>Portfolio project governance</b>	Alignment with organizational governance	How does analytics enable alignment of portfolio management with organizational governance?
	Defining portfolio decision-making hierarchy	What is the in defining the decision-making hierarchy in portfolio management for analytic projects?
	Defining portfolio governance roles	How portfolio governance roles may be defined in analytics/data science projects?
<b>Portfolio capacity management</b>	Defining organizational capabilities	How does analytics contribute to the identification and development of organizational capabilities in portfolio management?
	Capacity planning	How can analytics be used to improve capacity planning in portfolio management?

	Supply and demand optimization	What is the impact of analytics on supply and demand optimization in portfolio management?
<b>Portfolio stakeholder management</b>	Definition and identification of portfolio stakeholders	How does analytics enable the identification and engagement of portfolio stakeholders?
	Stakeholder engagement planning	How can analytics be used to improve stakeholder engagement planning in portfolio management?
	Portfolio communication management	What is the impact of analytics on portfolio communication management?
<b>Financial management</b>	Portfolio component cost estimations	How has the use of analytics improved the accuracy and reliability of cost estimations for portfolio components?
	Portfolio budget	How has analytics contributed to the optimization of budget allocation across portfolio projects?
	Financial management control	How has analytics enhanced financial management control within the portfolio?
<b>Value management</b>	Criteria for setting value objectives	How can analytics be used to establish criteria for setting value objectives in portfolio management?
	Maximizing value	How can analytics be used to maximize value in portfolio management?
	Monitoring and measuring value	How can analytics be used to monitor and measure value in portfolio management?
<b>Risk management</b>	Risk identification	How can analytics be used to identify and assess portfolio risks?
	Risk analysis	What is the impact of analytics on risk analysis in portfolio management?
	Risk response	How can analytics be used to improve risk response in portfolio management?
<b>Change Management</b>	Identification of changes	How has the use of analytics facilitated the identification of potential changes within the portfolio?
	Impact assessment	In what ways has analytics improved the evaluation and assessment of the potential impact of changes on the portfolio?
	Change control	How has analytics supported the effective management and control of approved changes within the portfolio?
<b>Portfolio monitoring and control</b>	Metrics for monitoring project portfolio	How can metrics for monitoring project portfolio be improved with analytics?
	Monitoring of capabilities and performance	How does the use of analytics enable the monitoring of capabilities and performance in portfolio management?
	Portfolio value reporting and analysis	What is the impact of analytics on portfolio value reporting and analysis?

	Monitoring of portfolio risks	How can analytics be used to improve the monitoring of portfolio risks in portfolio management?
--	-------------------------------	---

*Figura 2: Tabla de criterios y categorías del framework unificado que se usa para la etiquetación*

### 3. Formato de salida:

data/classifiedData: Se genera un CSV en formato UTF-8. Este contiene todos los artículos o proyectos ingresados en el archivo “dataToClassify” clasificados con el siguiente formato:

- Title (título del artículo o proyecto)
- Abstract (resumen del artículo o proyecto)
- Category (categoría asignada al artículo o proyecto)
- Criteria (criterio asignado al artículo o proyecto)

data/inputAffinity: Se genera un CSV en formato UTF-8. Este contiene todos los artículos o proyectos ingresados en el archivo “dataToClassify” junto a un puntaje de afinidad respecto al conjunto de palabras de cada categoría en columnas.

data/transformedFrameworkData: Se genera un CSV en formato UTF-8. Este contiene la información de todos los artículos verificados luego de ser transformados dentro del modelo en el siguiente formato:

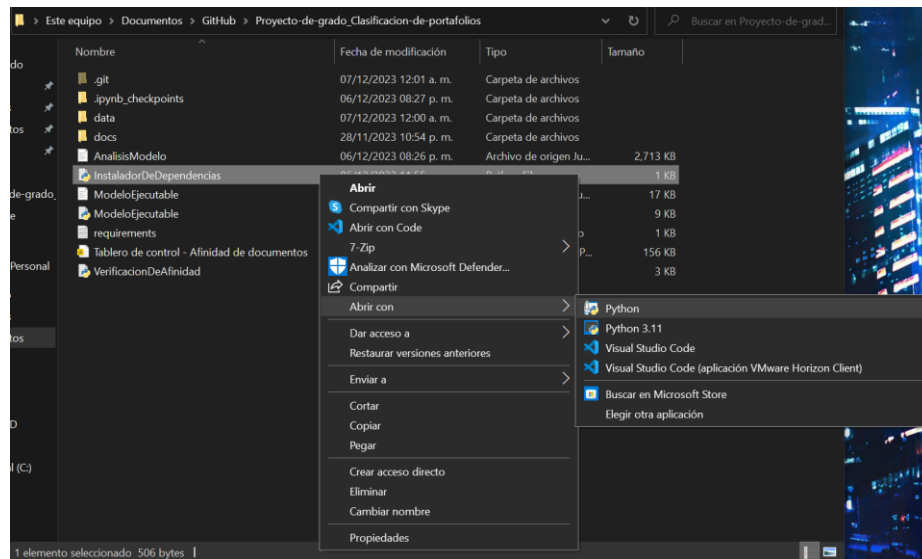
- Title (título del artículo o proyecto)
- Abstract (resumen del artículo o proyecto)
- Category (categoría asignada al artículo o proyecto)
- Words (palabras relevantes que se usaran para la clasificación)

data/transformedInputData: Se genera un CSV en formato UTF-8. Este contiene la información de todos los artículos ingresados en el archivo “dataToClassify” luego de ser transformados dentro del modelo en el siguiente formato:

- Title (título del artículo o proyecto)
- Abstract (resumen del artículo o proyecto)
- Words (palabras relevantes que se usaran para la clasificación)

### 4. Guía de ejecución:

- Para abrir los archivos .py, se puede oprimir clic izquierdo 2 veces o presionar clic derecho sobre el archivo y abrirlo seleccionado el interpretador de Python.



*Figura 3: Ejecución de un archivo .py a través del interpretador de Python*

1. Ejecute el archivo “InstaladorDeDependencias.py”. Este instalará en su equipo todas las librerías necesarias para correr el proyecto. Una ventana de cmd se abrirá mostrando el estado de la descarga de cada librería y se cerrará cuando todas hayan sido instaladas. Entre las librerías que se descargarán se encuentran nltk, pandas, numpy, seaborn, contractions, wordcloud, scikit-learn, imbalanced-learn y matplotlib. Solo es necesario ejecutarlo una vez. Si se corre de nuevo, la consola se cerrará más rápido mientras indica que las librerías ya fueron instaladas.
2. Dentro de la carpeta data, guarde el archivo de los artículos o proyectos que se van a clasificar como “dataToClassify” y el archivo de entrenamiento como “verifiedArticles”.
3. Ejecute el archivo “ModeloEjecutable.py”. Este realizará el proceso de lectura, transformación, vectorización y predicción con los 2 archivos provistos en el paso anterior. Luego de que se cierre la consola de cmd, se habrán generado dentro de la carpeta data los archivos “classifiedData”, “transformedFrameworkData” y “transformedInputData”.
4. (Opcional) Si desea generar el reporte de afinidad por categoría de cada artículo o proyecto, ejecute el archivo “VerificadorDeAfinidad.py”. Cuando la consola de cmd se cierre, se generará el reporte dentro de la carpeta data en un archivo llamado “inputAffinity”.

## 5. Uso de Biblioshiny

Si desea analizar estadísticas documentales de los archivos importados de Scopus, se recomienda el uso de Biblioshiny, la interfaz web de Bibliometrix. Esta permite transformar los archivos de entrada a dataframes, filtrar la información y generar estadísticas a nivel de fuentes, autores, documentos y agrupamiento por acoplamiento.

Luego de haber instalado R y RStudio, introduzca en orden los siguientes comandos en la consola de RStudio para ejecutar la interfaz:

- `install.packages("bibliometrix")`
- `library(bibliometrix)`

- biblioshiny()

Si tiene alguna duda sobre la instalación o quiere conocer a fondo la herramienta, vea esta guía detallada de Bibliometrix para ejecutar la interfaz y utilizar sus funciones: <https://bibliometrix.org/biblioshiny/biblioshiny1.html>

## 6. Problemas frecuentes:

En caso de que, al ejecutar los archivos .py, la consola de cmd se cierre inmediatamente y no se muestren cambios en el estado del directorio, puede realizar lo siguiente:

InstaladorDeDependencias.py: Revise si pip está instalado correctamente dentro del equipo y Python esta registrado dentro del PATH de las variables de entorno.

ModeloEjecutable.py: Revise el formato de los archivos “dataToClassify” y “verifiedArticles” especificado en la sección *formato de entrada*. Además, elimine filas vacías dentro de los archivos.

VerificadorDeAfinidad.py: Revise que los archivos “transformedFrameworkData” y “transformedInputData” hayan sido generados con el ejecutable del modelo.