



# Clasificación de Documentos

Daniel Rodríguez Aguilar

## ÍNDICE

JUSTIFICACIÓN DEL PROYECTO .....	2
TIPO DE TEXTOS Y CATEGORÍAS .....	3
TEXTOS DE ENTRENAMIENTO Y TEST .....	4
PALABRAS CLAVE .....	5
EXTRACCIÓN DE LAS CARACTERÍSTICAS DEL CONJUNTO DE ENTRENAMIENTO .....	6
IMPLEMENTACIÓN DE LOS CLASIFICADORES .....	7
REALIZACIÓN DE EXPERIMENTOS .....	8
MANUAL DE USUARIO .....	17
ANEXO .....	19

## JUSTIFICACIÓN DEL PROYECTO

El objetivo es el de clasificar documentos que pertenecen a un mismo grupo pero que a su vez se dividen en subcategorías de éste. Para ello hemos hecho uso de dos algoritmos: Naive Bayes y kNN.

El algoritmo de Naive Bayes se fundamenta en el teorema de Bayes y su clasificador asume que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de cualquier otra característica, dada la categoría variable.

Por otro lado, el algoritmo kNN estima la probabilidad a posteriori de que un elemento pertenezca a una determinada categoría a partir de la información proporcionada por el conjunto de entrenamiento clasificándose en la clase más frecuente a la que pertenecen sus K vecinos más cercanos.

## TIPO DE TEXTOS Y CATEGORÍAS

La aplicación que se ha llevado a cabo es la de clasificar noticias relacionadas con el deporte y se han considerado las siguientes categorías:

- Fútbol
- Baloncesto
- Tenis
- Fórmula 1
- Ciclismo

Se han elegido estas categorías ya que son fácilmente diferenciables, pero a su vez comparten términos que hacen que la clasificación no sea tan sencilla.

## TEXTOS DE ENTRENAMIENTO Y TEST

Se han seleccionado textos que comparten las siguientes características:

- **Lenguaje:** Español
- **Formato:** Texto plano

Para cada categoría se han recopilado la siguiente cantidad de documentos de entrenamiento:

- **Fútbol:** 20
- **Baloncesto:** 15
- **Tenis:** 15
- **Fórmula 1:** 15
- **Ciclismo:** 15

Para cada categoría se han recopilado la siguiente cantidad de documentos adicionales para testear:

- **Fútbol:** 2
- **Baloncesto:** 2
- **Tenis:** 2
- **Fórmula 1:** 2
- **Ciclismo:** 2

Todos los textos se han almacenado en su carpeta correspondiente (“entrenamiento” ó “adicionales”) en formato “.txt”. Para acceder a la fuente de cada texto se puede consultar el Anexo.

## PALABRAS CLAVE

Se han elegido manualmente 20 palabras clave por categoría, en las cuáles se repite una o más palabras con otra categoría debido a su relevancia.

- **Fútbol:** ["futbol", "partido", "equipo", "resultado", "remate", "amarilla", "jugador", "minuto", "gol", "victoria", "derrota", "empate", "punto", "falta", "cabeza", "penalti", "portero", "defensa", "jornada", "balon"]
- **Baloncesto:** ["baloncesto", "alero", "doble-doble", "triple", "campo", "jugador", "asistencia", "partido", "rebote", "punto", "victoria", "derrota", "pivot", "cuarto", "personal", "segundo", "libre", "tiro", "base", "escolta"]
- **Tenis:** ["tenis", "partido", "masters", "slam", "ranking", "set", "tenista", "punto", "torneo", "victoria", "titulo", "atp", "juego", "manga", "dobles", "minuto", "mundo", "numero", "final", "saque"]
- **Fórmula 1:** ["mercedes", "mclaren", "ferrari", "bull", "renault", "victoria", "equipo", "escuderia", "carrera", "circuito", "f1", "vuelta", "piloto", "motor", "rueda", "temporada", "puesto", "pole", "curva", "coche"]
- **Ciclismo:** ["ciclista", "bici", "tour", "vuelta", "caida", "corredor", "etapa", "giro", "equipo", "meta", "kilometro", "esprint", "victoria", "circuito", "ruta", "recorrido", "italia", "españa", "francia", "puesto"]

## EXTRACCIÓN DE LAS CARACTERÍSTICAS DEL CONJUNTO DE ENTRENAMIENTO

Para el procesado de textos se ha hecho uso del *kit de herramientas de lenguaje natural*, o más comúnmente conocido como NLTK. Este kit contiene un conjunto de bibliotecas y programas para procesamiento de lenguaje natural en Python.

Para ambos algoritmos se ha usado NLTK para obtener las raíces de las palabras que aparecen más de una vez para evitar sus plurales y diferentes formas en las que puede aparecer una determinada palabra con la misma raíz. Así como para eliminar del conjunto de palabras a analizar aquellas determinadas como “palabras vacías” o “stopwords”, cuyo significado y relevancia en un texto es nulo.

Para el entrenamiento mediante kNN, los resultados obtenidos se almacenan en un documento de texto llamado “Entrenamiento kNN.txt”, en el cuál se muestra:

- El orden de las palabras que han sido analizadas
- La frecuencia documental inversa de cada una de ellas
- La representación vectorial de cada uno de los textos de entrenamiento con los pesos de cada término en ellos

Para el entrenamiento mediante Naive Bayes, los resultados obtenidos se almacenan en un documento de texto llamado “Entrenamiento Naive Bayes.txt”, en el cuál se muestra:

- El orden de las palabras que han sido analizadas
- Los valores  $P(c)$  y  $P(t|c)$  para cada una de las categorías existentes y términos del vocabulario

## IMPLEMENTACIÓN DE LOS CLASIFICADORES

### Clasificador kNN

La función que implementa este clasificador hace uso de los resultados del entrenamiento, obtenidos de “Entrenamiento kNN.txt”, así como del nuevo documento a clasificar y un valor ‘k’.

Al no ser clasificación binaria se ha implementado una variante del kNN que suma la similitud de los documentos de entrenamiento de la misma categoría más cercanos al nuevo, dado ‘k’. El resultado devuelve la categoría cuyo valor de la suma es mayor.

### Clasificador Naive Bayes

La función que implementa este clasificador hace uso de los resultados del entrenamiento, obtenidos de “Entrenamiento Naive Bayes.txt”, así como del nuevo documento a clasificar.

El resultado devuelve la categoría cuya probabilidad de ser la correcta es mayor.

***Nota:*** el resultado de las clasificaciones se muestra por consola y no se almacena en ningún archivo.

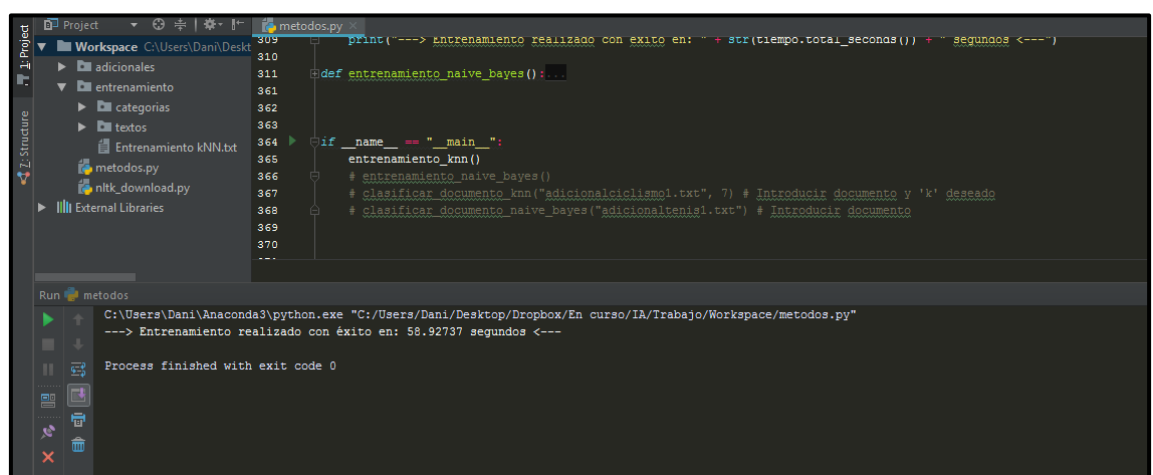


## REALIZACIÓN DE EXPERIMENTOS

En este apartado se van a comparar los resultados de los distintos clasificadores. También se realizará una comparativa en cuanto a rendimiento para ambos procesos de entrenamiento y clasificación.

### Entrenamiento

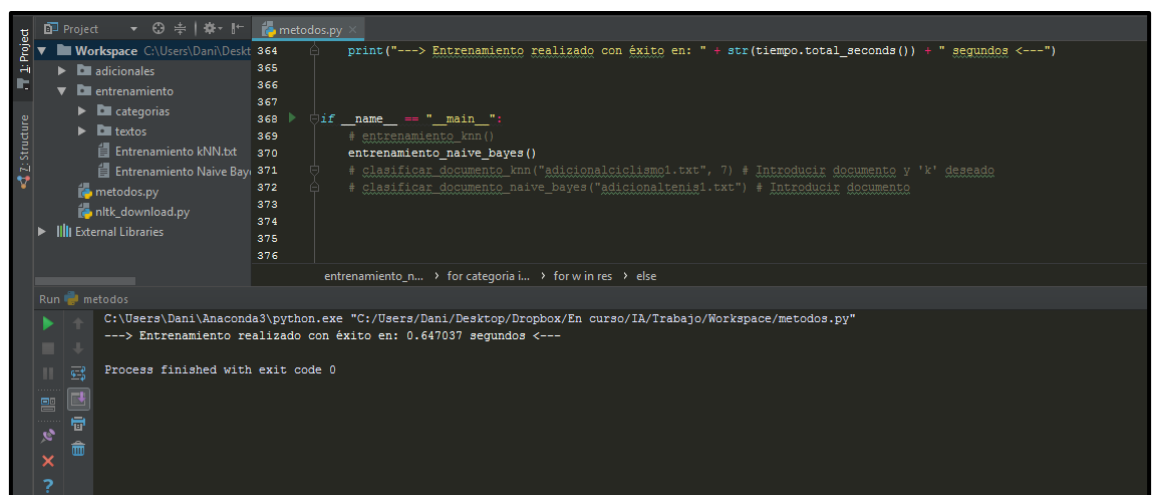
- **kNN:** 58.92737 segundos



```
309 print("----> Entrenamiento realizado con éxito en: " + str(tiempo.total_seconds()) + " segundos <---")
310
311 def entrenamiento_naive_bayes(): ...
312
313
314 if __name__ == "__main__":
315     entrenamiento_knn()
316     # entrenamiento_naive_bayes()
317     # clasificar_documento_knn("adicionalciclismo1.txt", 7) # Introducir documento y 'k' deseado
318     # clasificar_documento_naive_bayes("adicionaltenis1.txt") # Introducir documento
319
320
321
```

Run: C:\Users\Dani\Anaconda3\python.exe "C:\Users\Dani\Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py"  
----> Entrenamiento realizado con éxito en: 58.92737 segundos <---  
Process finished with exit code 0

- **Naive Bayes:** 0.647037 segundos



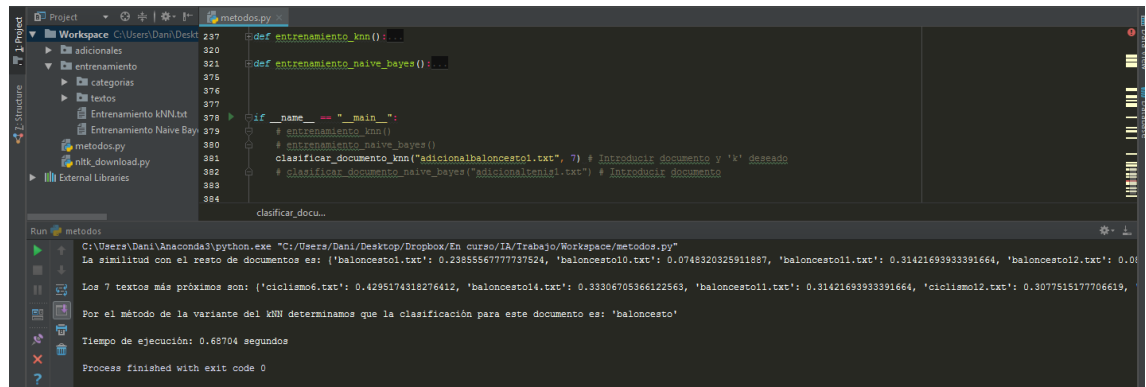
```
364 print("----> Entrenamiento realizado con éxito en: " + str(tiempo.total_seconds()) + " segundos <---")
365
366
367
368 if __name__ == "__main__":
369     # entrenamiento_knn()
370     entrenamiento_naive_bayes()
371     # clasificar_documento_knn("adicionalciclismo1.txt", 7) # Introducir documento y 'k' deseado
372     # clasificar_documento_naive_bayes("adicionaltenis1.txt") # Introducir documento
373
374
375
376
```

Run: C:\Users\Dani\Anaconda3\python.exe "C:\Users\Dani\Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py"  
----> Entrenamiento realizado con éxito en: 0.647037 segundos <---  
Process finished with exit code 0

## Clasificador

### ➤ KNN

- “adicionalbaloncesto1”



```
def entrenamiento_knn():
def entrenamiento_naive_bayes():

if __name__ == "__main__":
    # entrenamiento_knn()
    # entrenamiento_naive_bayes()
    clasificar_documento_knn("adicionalbaloncesto1.txt", 7) # Introducir documento y 'k' deseado
    # clasificar_documento_naive_bayes("adicionaltenis1.txt") # Introducir documento

clasificar_docu...
```

Run metodos

C:\Users\Dani\Anaconda3\python.exe "C:\Users\Dani\Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py"

La similitud con el resto de documentos es: {'baloncesto1.txt': 0.23855677777777778, 'baloncesto10.txt': 0.0748320325911887, 'baloncesto11.txt': 0.31421693933391664, 'baloncesto12.txt': 0.0...

Los 7 textos más próximos son: {'ciolismo6.txt': 0.4295174318276412, 'baloncesto14.txt': 0.33306705366122563, 'baloncesto11.txt': 0.31421693933391664, 'ciolismo12.txt': 0.3077515177706619, ...}

Por el método de la variante del kNN determinamos que la clasificación para este documento es: 'baloncesto'

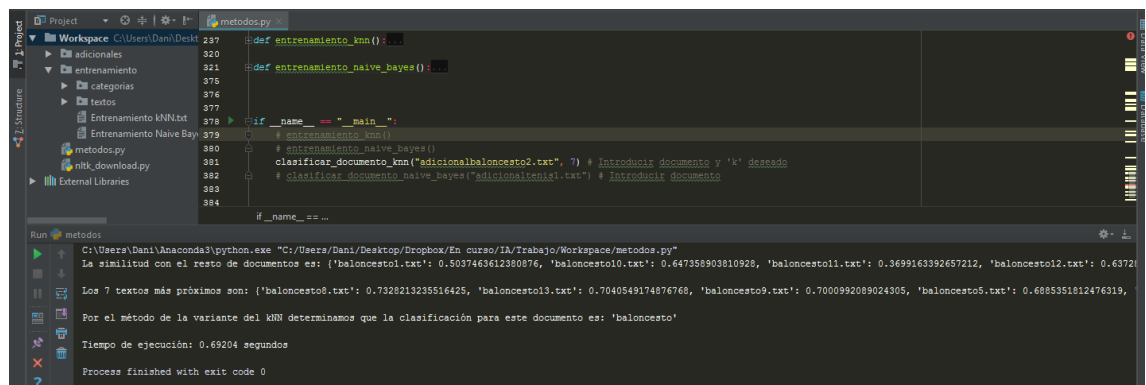
Tiempo de ejecución: 0.68704 segundos

Process finished with exit code 0

*Clasificación:* ‘baloncesto’

*Tiempo de ejecución:* 0.68704 segundos

- “adicionalbaloncesto2”



```
def entrenamiento_knn():
def entrenamiento_naive_bayes():

if __name__ == "__main__":
    # entrenamiento_knn()
    # entrenamiento_naive_bayes()
    clasificar_documento_knn("adicionalbaloncesto2.txt", 7) # Introducir documento y 'k' deseado
    # clasificar_documento_naive_bayes("adicionaltenis1.txt") # Introducir documento

# _name__ == ...
```

Run metodos

C:\Users\Dani\Anaconda3\python.exe "C:\Users\Dani\Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py"

La similitud con el resto de documentos es: {'baloncesto1.txt': 0.5037463612380876, 'baloncesto10.txt': 0.647358903810928, 'baloncesto11.txt': 0.3699163392657212, 'baloncesto12.txt': 0.6372...

Los 7 textos más próximos son: {'baloncesto8.txt': 0.7328213235516425, 'baloncesto13.txt': 0.7040549174876768, 'baloncesto9.txt': 0.7000992089024305, 'baloncesto5.txt': 0.6885351812476319, ...}

Por el método de la variante del kNN determinamos que la clasificación para este documento es: 'baloncesto'

Tiempo de ejecución: 0.69204 segundos

Process finished with exit code 0

*Clasificación:* ‘baloncesto’

*Tiempo de ejecución:* 0.69204 segundos

- “adicionalciclismo1”

```

def entrenamiento_knn():
def entrenamiento_naive_bayes():

if __name__ == "__main__":
    # entrenamiento_knn()
    # entrenamiento_naive_bayes()
    clasificar_documento_knn("adicionalciclismo1.txt", 7) # Introducir documento y 'k' deseado
    # clasificar_documento_naive_bayes("adicionalciclismo1.txt") # Introducir documento

if __name__ == __

```

Run metodos

C:\Users\Dani\Anaconda3\python.exe "C:\Users\Dani\Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py"

La similitud con el resto de documentos es: ('baloncesto1.txt': 0.0, 'baloncesto10.txt': 0.0, 'baloncesto11.txt': 0.0, 'baloncesto12.txt': 0.05337013701232462, 'baloncesto13.txt': 0.0, 'bal'

Los 7 textos más próximos son: ('ciclismo9.txt': 0.6865021608212796, 'ciclismo7.txt': 0.6286981586301706, 'ciclismo13.txt': 0.5056703435303485, 'ciclismo15.txt': 0.3503233485673246, 'formul'

Por el método de la variante del KNN determinamos que la clasificación para este documento es: 'ciclismo'

Tiempo de ejecución: 0.838047 segundos

Process finished with exit code 0

*Clasificación: ‘ciclismo’*

*Tiempo de ejecución: 0.838047 segundos*

- “adicionalciclismo2”

```

def entrenamiento_knn():
def entrenamiento_naive_bayes():

if __name__ == "__main__":
    # entrenamiento_knn()
    # entrenamiento_naive_bayes()
    clasificar_documento_knn("adicionalciclismo2.txt", 7) # Introducir documento y 'k' deseado
    # clasificar_documento_naive_bayes("adicionalciclismo2.txt") # Introducir documento

if __name__ == __

```

Run metodos

C:\Users\Dani\Anaconda3\python.exe "C:\Users\Dani\Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py"

La similitud con el resto de documentos es: ('baloncesto1.txt': 0.0, 'baloncesto10.txt': 0.006031398788597062, 'baloncesto11.txt': 0.0161222410718673, 'baloncesto12.txt': 0.0, 'baloncesto13'

Los 7 textos más próximos son: ('ciclismo8.txt': 0.887353301969747, 'ciclismo1.txt': 0.710882884858925, 'ciclismo2.txt': 0.6768328027439001, 'ciclismo6.txt': 0.668125787074877, 'ciclismo8'

Por el método de la variante del KNN determinamos que la clasificación para este documento es: 'ciclismo'

Tiempo de ejecución: 0.968055 segundos

Process finished with exit code 0

*Clasificación: ‘ciclismo’*

*Tiempo de ejecución: 0.968055 segundos*

- “adicionalformula1”

```

def entrenamiento_knn():
def entrenamiento_naive_bayes():

if __name__ == "__main__":
    # entrenamiento_knn()
    # entrenamiento_naive_bayes()
    clasificar_documento_knn("adicionalformula1.txt", 7) # Introducir documento y 'k' deseado
    # clasificar_documento_naive_bayes("adicionalformula1.txt") # Introducir documento

if __name__ == __

```

Run metodos

C:\Users\Dani\Anaconda3\python.exe "C:\Users\Dani\Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py"

La similitud con el resto de documentos es: ('baloncesto1.txt': 0.06091836306764161, 'baloncesto10.txt': 0.04489575855760434, 'baloncesto11.txt': 0.04273715413290985, 'baloncesto12.txt': 0.0

Los 7 textos más próximos son: ('formula11.txt': 0.7004855008109999, 'formula13.txt': 0.6292244520799712, 'formula9.txt': 0.5509210244254659, 'formula15.txt': 0.48878467666091685, 'formula5'

Por el método de la variante del KNN determinamos que la clasificación para este documento es: 'formula'

Tiempo de ejecución: 0.743043 segundos

Process finished with exit code 0

*Clasificación: ‘formula’*

*Tiempo de ejecución: 0.743043 segundos*

- “adicionalformula2”

```

def entrenamiento_knn():
def entrenamiento_naive_bayes():

if __name__ == "__main__":
    # entrenamiento_knn()
    # entrenamiento_naive_bayes()
    clasificar_documento_knn("adicionalformula2.txt", 7) # Introducir documento y 'k' deseado
    # clasificar_documento_naive_bayes("adicionalformula1.txt") # Introducir documento

if __name__ == __

```

Run metodos

C:\Users\Dani\Anaconda3\python.exe "C:\Users\Dani\Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py"

La similitud con el resto de documentos es: {'baloncesto1.txt': 0.0, 'baloncesto10.txt': 0.018105878734538998, 'baloncesto11.txt': 0.0074373910362108605, 'baloncesto12.txt': 0.0061680598232...

Los 7 textos más próximos son: {'formula2.txt': 0.8152405641363241, 'formula15.txt': 0.6046996647841106, 'formula3.txt': 0.4197978183590077, 'formula11.txt': 0.26224115681330856, 'formula14...

Por el método de la variante del KNN determinamos que la clasificación para este documento es: 'formula'

Tiempo de ejecución: 0.449026 segundos

Process finished with exit code 0

*Clasificación: 'formula'*

*Tiempo de ejecución: 0.449026 segundos*

- “adicionalfutbol1”

```

def entrenamiento_knn():
def entrenamiento_naive_bayes():

if __name__ == "__main__":
    # entrenamiento_knn()
    # entrenamiento_naive_bayes()
    clasificar_documento_knn("adicionalfutbol1.txt", 7) # Introducir documento y 'k' deseado
    # clasificar_documento_naive_bayes("adicionalformula1.txt") # Introducir documento

if __name__ == __

```

Run metodos

C:\Users\Dani\Anaconda3\python.exe "C:\Users\Dani\Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py"

La similitud con el resto de documentos es: {'baloncesto1.txt': 0.04706317939386265, 'baloncesto10.txt': 0.09963858118376542, 'baloncesto11.txt': 0.16251970881241434, 'baloncesto12.txt': 0.0...

Los 7 textos más próximos son: {'futbol9.txt': 0.8013542871130672, 'futbol11.txt': 0.599792890525534, 'futbol6.txt': 0.5778969155156676, 'futbol19.txt': 0.5252318662933367, 'futbol11.txt': 0.5...

Por el método de la variante del KNN determinamos que la clasificación para este documento es: 'futbol'

Tiempo de ejecución: 0.70404 segundos

Process finished with exit code 0

*Clasificación: 'futbol'*

*Tiempo de ejecución: 0.70404 segundos*

- “adicionalfutbol2”

```

def entrenamiento_knn():
def entrenamiento_naive_bayes():

if __name__ == "__main__":
    # entrenamiento_knn()
    # entrenamiento_naive_bayes()
    clasificar_documento_knn("adicionalfutbol2.txt", 7) # Introducir documento y 'k' deseado
    # clasificar_documento_naive_bayes("adicionalformula1.txt") # Introducir documento

if __name__ == __

```

Run metodos

C:\Users\Dani\Anaconda3\python.exe "C:\Users\Dani\Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py"

La similitud con el resto de documentos es: {'baloncesto1.txt': 0.13186092034228428, 'baloncesto10.txt': 0.20212217458326728, 'baloncesto11.txt': 0.4960881166136745, 'baloncesto12.txt': 0.1...

Los 7 textos más próximos son: {'futbol16.txt': 0.6720602680570398, 'futbol7.txt': 0.5359361076545155, 'baloncesto11.txt': 0.4960881166136745, 'futbol2.txt': 0.46473696825005234, 'futbol11.txt': 0.5...

Por el método de la variante del KNN determinamos que la clasificación para este documento es: 'futbol'

Tiempo de ejecución: 0.592034 segundos

Process finished with exit code 0

*Clasificación: 'futbol'*

*Tiempo de ejecución: 0.592034 segundos*

- *“adicionaltenis1”*

The screenshot displays a Jupyter Notebook environment with a file explorer on the left showing the project structure. The main area contains a Python script with the following code:

```
def entrenamiento_knn():
def entrenamiento_naive_bayes():
if __name__ == "__main__":
    # entrenamiento_knn()
    # entrenamiento_naive_bayes()
    clasificar_documento_knn("adicionaltenis1.txt", 7) # Introducir documento y 'k' deseado
    # clasificar_documento_naive_bayes("adicionaltenis1.txt") # Introducir documento
```

The output of the script is displayed below the code cell, showing the results of the KNN classification for a set of documents. It lists the most similar documents and the predicted class for a new document.

Los 7 textos más próximos son: ['tenis11.txt': 0.4460147988953074, 'tenis5.txt': 0.3998014874359835, 'tenis4.txt': 0.3896759304101031, 'tenis12.txt': 0.3717672706482086, 'tenis9.txt': 0.333...

Por el método de la variante del KNN determinamos que la clasificación para este documento es: 'tenis'

Tiempo de ejecución: 0.35802 segundos

Process finished with exit code 0

*Clasificación:* 'tenis'

*Tiempo de ejecución:* 0.35802 segundos

- “*adicionaltenis2*”

The screenshot displays a Jupyter Notebook environment with the following components:

- File Explorer (Left):** Shows the project structure, including a workspace folder containing files like 'entrenamiento\_knn.py' and 'entrenamiento\_naive\_bayes.py'.
- Code Editor (Center):** Contains the Python script for KNN classification. The script defines functions for training and classifying documents based on word frequency.
- Run Output (Bottom):** Shows the execution results of the script, including the path to the notebook file and the classification results for a test document.

**Python Script Content:**

```
def entrenamiento_knn():
    # ... (code for training the KNN model) ...

def entrenamiento_naive_bayes():
    # ... (code for training the Naive Bayes model) ...

if __name__ == "__main__":
    # ... (main execution logic) ...
```

**Run Output:**

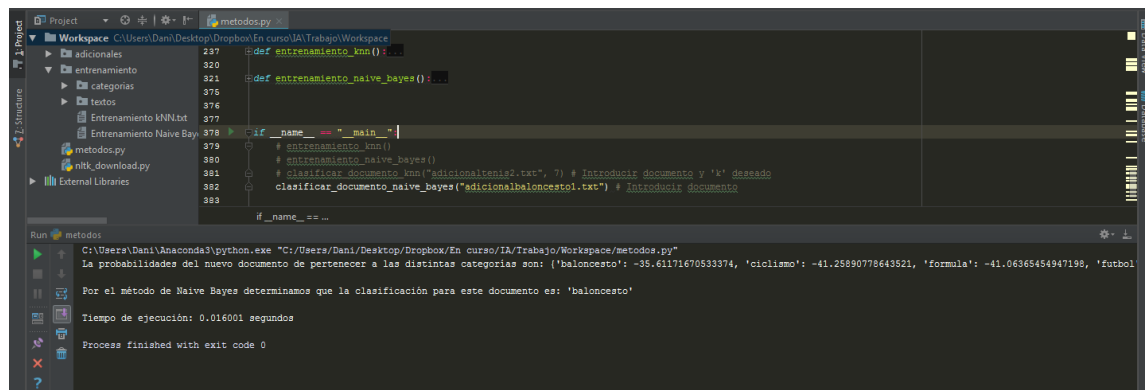
```
C:\Users\Dani\Anaconda3\python.exe "C:/Users/Dani/Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py"
La similitud con el resto de documentos es: {'baloncesto1.txt': 0.12977973986830516, 'baloncesto10.txt': 0.025749347613683692, 'baloncesto11.txt': 0.08729576593742736, 'baloncesto12.txt': 0.04511111111111111, 'baloncesto13.txt': 0.04511111111111111, 'baloncesto14.txt': 0.04511111111111111, 'baloncesto15.txt': 0.04511111111111111, 'baloncesto16.txt': 0.04511111111111111, 'baloncesto17.txt': 0.04511111111111111, 'baloncesto18.txt': 0.04511111111111111, 'baloncesto19.txt': 0.04511111111111111, 'baloncesto2.txt': 0.04511111111111111, 'baloncesto3.txt': 0.04511111111111111, 'baloncesto4.txt': 0.04511111111111111, 'baloncesto5.txt': 0.04511111111111111, 'baloncesto6.txt': 0.04511111111111111, 'baloncesto7.txt': 0.04511111111111111, 'baloncesto8.txt': 0.04511111111111111, 'baloncesto9.txt': 0.04511111111111111, 'tenis1.txt': 0.6591431512423383, 'tenis2.txt': 0.6591431512423383, 'tenis3.txt': 0.6451111111111111, 'tenis4.txt': 0.6451111111111111, 'tenis5.txt': 0.6451111111111111, 'tenis6.txt': 0.6451111111111111, 'tenis7.txt': 0.6451111111111111, 'tenis8.txt': 0.6451111111111111, 'tenis9.txt': 0.6451111111111111, 'tenis10.txt': 0.6451111111111111, 'tenis11.txt': 0.6451111111111111, 'tenis12.txt': 0.6451111111111111, 'tenis13.txt': 0.6451111111111111, 'tenis14.txt': 0.6451111111111111, 'tenis15.txt': 0.6451111111111111, 'tenis16.txt': 0.6451111111111111, 'tenis17.txt': 0.6451111111111111, 'tenis18.txt': 0.6451111111111111, 'tenis19.txt': 0.6451111111111111, 'tenis20.txt': 0.6451111111111111, 'tenis21.txt': 0.6451111111111111, 'tenis22.txt': 0.6451111111111111, 'tenis23.txt': 0.6451111111111111, 'tenis24.txt': 0.6451111111111111, 'tenis25.txt': 0.6451111111111111, 'tenis26.txt': 0.6451111111111111, 'tenis27.txt': 0.6451111111111111, 'tenis28.txt': 0.6451111111111111, 'tenis29.txt': 0.6451111111111111, 'tenis30.txt': 0.6451111111111111, 'tenis31.txt': 0.6451111111111111, 'tenis32.txt': 0.6451111111111111, 'tenis33.txt': 0.6451111111111111, 'tenis34.txt': 0.6451111111111111, 'tenis35.txt': 0.6451111111111111, 'tenis36.txt': 0.6451111111111111, 'tenis37.txt': 0.6451111111111111, 'tenis38.txt': 0.6451111111111111, 'tenis39.txt': 0.6451111111111111, 'tenis40.txt': 0.6451111111111111, 'tenis41.txt': 0.6451111111111111, 'tenis42.txt': 0.6451111111111111, 'tenis43.txt': 0.6451111111111111, 'tenis44.txt': 0.6451111111111111, 'tenis45.txt': 0.6451111111111111, 'tenis46.txt': 0.6451111111111111, 'tenis47.txt': 0.6451111111111111, 'tenis48.txt': 0.6451111111111111, 'tenis49.txt': 0.6451111111111111, 'tenis50.txt': 0.6451111111111111, 'tenis51.txt': 0.6451111111111111, 'tenis52.txt': 0.6451111111111111, 'tenis53.txt': 0.6451111111111111, 'tenis54.txt': 0.6451111111111111, 'tenis55.txt': 0.6451111111111111, 'tenis56.txt': 0.6451111111111111, 'tenis57.txt': 0.6451111111111111, 'tenis58.txt': 0.6451111111111111, 'tenis59.txt': 0.6451111111111111, 'tenis60.txt': 0.6451111111111111, 'tenis61.txt': 0.6451111111111111, 'tenis62.txt': 0.6451111111111111, 'tenis63.txt': 0.6451111111111111, 'tenis64.txt': 0.6451111111111111, 'tenis65.txt': 0.6451111111111111, 'tenis66.txt': 0.6451111111111111, 'tenis67.txt': 0.6451111111111111, 'tenis68.txt': 0.6451111111111111, 'tenis69.txt': 0.6451111111111111, 'tenis70.txt': 0.6451111111111111, 'tenis71.txt': 0.6451111111111111, 'tenis72.txt': 0.6451111111111111, 'tenis73.txt': 0.6451111111111111, 'tenis74.txt': 0.6451111111111111, 'tenis75.txt': 0.6451111111111111, 'tenis76.txt': 0.6451111111111111, 'tenis77.txt': 0.6451111111111111, 'tenis78.txt': 0.6451111111111111, 'tenis79.txt': 0.6451111111111111, 'tenis80.txt': 0.6451111111111111, 'tenis81.txt': 0.6451111111111111, 'tenis82.txt': 0.6451111111111111, 'tenis83.txt': 0.6451111111111111, 'tenis84.txt': 0.6451111111111111, 'tenis85.txt': 0.6451111111111111, 'tenis86.txt': 0.6451111111111111, 'tenis87.txt': 0.6451111111111111, 'tenis88.txt': 0.6451111111111111, 'tenis89.txt': 0.6451111111111111, 'tenis90.txt': 0.6451111111111111, 'tenis91.txt': 0.6451111111111111, 'tenis92.txt': 0.6451111111111111, 'tenis93.txt': 0.6451111111111111, 'tenis94.txt': 0.6451111111111111, 'tenis95.txt': 0.6451111111111111, 'tenis96.txt': 0.6451111111111111, 'tenis97.txt': 0.6451111111111111, 'tenis98.txt': 0.6451111111111111, 'tenis99.txt': 0.6451111111111111, 'tenis100.txt': 0.6451111111111111, 'tenis101.txt': 0.6451111111111111, 'tenis102.txt': 0.6451111111111111, 'tenis103.txt': 0.6451111111111111, 'tenis104.txt': 0.6451111111111111, 'tenis105.txt': 0.6451111111111111, 'tenis106.txt': 0.6451111111111111, 'tenis107.txt': 0.6451111111111111, 'tenis108.txt': 0.6451111111111111, 'tenis109.txt': 0.6451111111111111, 'tenis110.txt': 0.6451111111111111, 'tenis111.txt': 0.6451111111111111, 'tenis112.txt': 0.6451111111111111, 'tenis113.txt': 0.6451111111111111, 'tenis114.txt': 0.6451111111111111, 'tenis115.txt': 0.6451111111111111, 'tenis116.txt': 0.6451111111111111, 'tenis117.txt': 0.6451111111111111, 'tenis118.txt': 0.6451111111111111, 'ten
```

*Clasificación:* 'tenis'

*Tiempo de ejecución:* 1.153066 segundos

## ➤ Naive Bayes

- “adicionalbaloncesto1”



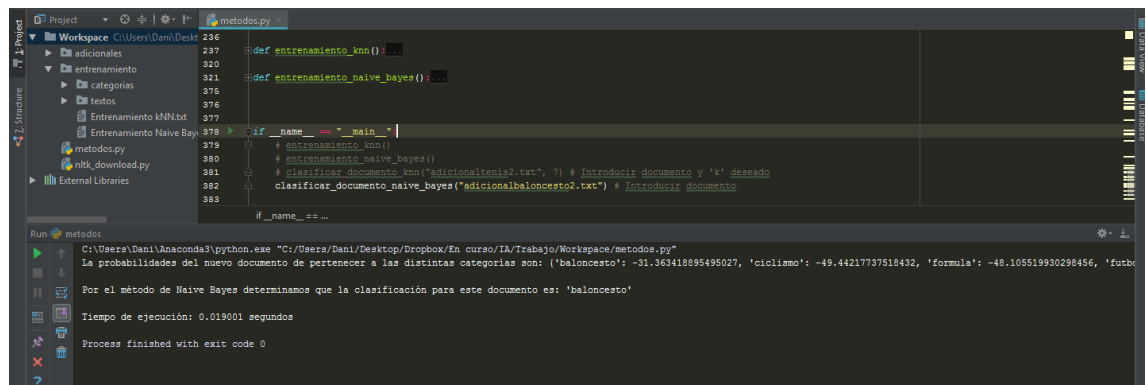
```
Project: metodos.py
Workspace: C:\Users\Dani\Desktop\Dropbox\En curso\IA\Trabajo\Workspace
227 def entrenamiento_knn():
320
321 def entrenamiento_naive_bayes():
375
376
377
378 if __name__ == "__main__":
379     # entrenamiento_knn()
380     # entrenamiento_naive_bayes()
381     # clasificar_documento_knn("adicionaltenis2.txt", 7) # Introducir documento y 'k' deseado
382     clasificar_documento_naive_bayes("adicionalbaloncesto1.txt") # Introducir documento
383
384 if __name__ == ...

Run: metodos
C:\Users\Dani\Anaconda3\python.exe "C:\Users\Dani\Desktop\Dropbox\En curso\IA\Trabajo\Workspace\metodos.py"
La probabilidades del nuevo documento de pertenecer a las distintas categorias son: ('baloncesto': -35.61171670533374, 'ciclismo': -41.25890778643521, 'formula': -41.06365454947198, 'futbol': -41.06365454947198)
Por el método de Naive Bayes determinamos que la clasificación para este documento es: 'baloncesto'
Tiempo de ejecución: 0.016001 segundos
Process finished with exit code 0
```

*Clasificación:* ‘baloncesto’

*Tiempo de ejecución:* 0.016001 segundos

- “adicionalbaloncesto2”



```
Project: metodos.py
Workspace: C:\Users\Dani\Desktop\Dropbox\En curso\IA\Trabajo\Workspace
226 def entrenamiento_knn():
227
228 def entrenamiento_naive_bayes():
275
276
277
278 if __name__ == "__main__":
279     # entrenamiento_knn()
280     # entrenamiento_naive_bayes()
281     # clasificar_documento_knn("adicionaltenis2.txt", 7) # Introducir documento y 'k' deseado
282     clasificar_documento_naive_bayes("adicionalbaloncesto2.txt") # Introducir documento
283
284 if __name__ == ...

Run: metodos
C:\Users\Dani\Anaconda3\python.exe "C:\Users\Dani\Desktop\Dropbox\En curso\IA\Trabajo\Workspace\metodos.py"
La probabilidades del nuevo documento de pertenecer a las distintas categorias son: ('baloncesto': -31.363418895495027, 'ciclismo': -49.44217737518432, 'formula': -48.105519930298456, 'futbol': -48.105519930298456)
Por el método de Naive Bayes determinamos que la clasificación para este documento es: 'baloncesto'
Tiempo de ejecución: 0.019001 segundos
Process finished with exit code 0
```

*Clasificación:* ‘baloncesto’

*Tiempo de ejecución:* 0.019001 segundos

- “adicionalciclismo1”

The screenshot displays a Jupyter Notebook environment. The left sidebar shows the file explorer with a project named 'Project' containing a 'Workspace' folder. The main area shows a Python script named 'metodos.py' with the following code:

```
def entrenamiento_knn(): ...
def entrenamiento_naive_bayes(): ...
if __name__ == "__main__":
    # entrenamiento_knn()
    # entrenamiento_naive_bayes()
    # clasificar_documento_knn("adicionaltenis7.txt", 7) # Introducir documento y 'k' deseado
    clasificar_documento_naive_bayes("adicionalciclismo1.txt") # Introducir documento
if __name__ == ...
```

The output of the script is displayed below the code cell:

```
Run
C:\Users\Dani\Anaconda3\python.exe C:\Users/Dani/Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py
La probabilidades del nuevo documento de pertenecer a las distintas categorias son: {'baloncesto': -32.81719673271807, 'ciclismo': -25.28920174563015, 'formula': -28.71695899737172, 'futbol': ...}
Por el metodo de Naive Bayes determinamos que la clasificacion para este documento es: 'ciclismo'
Tiempo de ejecucion: 0.020001 segundos
Process finished with exit code 0
```

*Clasificación: 'ciclismo'*

*Tiempo de ejecución:* 0.020001 segundos

- “*adicionalciclismo2*”

[illegible]

*Clasificación: 'ciclismo'*

*Tiempo de ejecución:* 0.023001 segundos

- “*adicionalformula1*”

The screenshot displays a Jupyter Notebook environment with a Python script for Naive Bayes classification. The script is named `metodos.py` and is located in the `Workspace` directory. The script defines two main functions: `entrenamiento_knn()` and `entrenamiento_naive_bayes()`. The `entrenamiento_naive_bayes()` function is currently selected, showing its implementation. The script also includes a `__main__` block that calls the `entrenamiento_naive_bayes()` function and prints the results.

```
def entrenamiento_knn():
def entrenamiento_naive_bayes():
if __name__ == "__main__":
    entrenamiento_knn()
    entrenamiento_naive_bayes()
    clasificador_documento_knn("adicionaltema2.txt", 7) + "Introducir documento y 'k' deseado
    clasificador_documento_naive_bayes("adicionalformula1.txt") + "Introducir documento
if __name__ == ...
```

The output of the script is displayed in the `Run` panel, showing the probabilities for a new document belonging to different categories and the final classification result.

```
C:\Users\Dani\Anaconda3\python.exe "C:/Users/Dani/Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py"
La probabilidades del nuevo documento de pertenecer a las distintas categorias son: {'baloncesto': -53.774626038165922, 'ciclismo': -47.690646421565766, 'formula': -40.3305339185494, 'furbu.
Por el método de Naive Bayes determinamos que la clasificación para este documento es: 'formula'
Tiempo de ejecución: 0.019001 segundos
Process finished with exit code 0
```

*Clasificación:* 'formula'

*Tiempo de ejecución:* 0.019001 segundos

- “adicionalformula2”

```

def entrenamiento_knn():
def entrenamiento_naive_bayes():
if __name__ == "__main__":
    # entrenamiento_knn()
    # entrenamiento_naive_bayes()
    # clasificar_documento_knn("adicionalformula2.txt", 7) # Introducir documento y 'k' deseado
    clasificar_documento_naive_bayes("adicionalformula2.txt") # Introducir documento
    # _name_ == __

```

Run metodos

C:\Users\Dani\Anaconda3\python.exe "C:/Users/Dani/Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py"

La probabilidades del nuevo documento de pertenecer a las distintas categorias son: ('baloncesto': -13.993236484713035, 'ciclismo': -13.743989127736185, 'formula': -9.66712833859393, 'futbol': -12.98523769513266)

Por el método de Naive Bayes determinamos que la clasificación para este documento es: 'formula'

Tiempo de ejecución: 0.016001 segundos

Process finished with exit code 0

*Clasificación: 'formula'*

*Tiempo de ejecución: 0.016001 segundos*

- “adicionalfutbol1”

```

def entrenamiento_knn():
def entrenamiento_naive_bayes():
if __name__ == "__main__":
    # entrenamiento_knn()
    # entrenamiento_naive_bayes()
    # clasificar_documento_knn("adicionalfutbol1.txt", 7) # Introducir documento y 'k' deseado
    clasificar_documento_naive_bayes("adicionalfutbol1.txt") # Introducir documento
    # _name_ == __

```

Run metodos

C:\Users\Dani\Anaconda3\python.exe "C:/Users/Dani/Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py"

La probabilidades del nuevo documento de pertenecer a las distintas categorias son: ('baloncesto': -28.211225495723323, 'ciclismo': -36.65071202178486, 'formula': -33.44570344514887, 'futbol': -17.68100578421954)

Por el método de Naive Bayes determinamos que la clasificación para este documento es: 'futbol'

Tiempo de ejecución: 0.019001 segundos

Process finished with exit code 0

*Clasificación: 'futbol'*

*Tiempo de ejecución: 0.019001 segundos*

- “adicionalfutbol2”

```

def entrenamiento_knn():
def entrenamiento_naive_bayes():
if __name__ == "__main__":
    # entrenamiento_knn()
    # entrenamiento_naive_bayes()
    # clasificar_documento_knn("adicionalfutbol2.txt", 7) # Introducir documento y 'k' deseado
    clasificar_documento_naive_bayes("adicionalfutbol2.txt") # Introducir documento
    # _name_ == __

```

Run metodos

C:\Users\Dani\Anaconda3\python.exe "C:/Users/Dani/Desktop/Dropbox/En curso/IA/Trabajo/Workspace/metodos.py"

La probabilidades del nuevo documento de pertenecer a las distintas categorias son: ('baloncesto': -12.98523769513266, 'ciclismo': -18.4975752101112, 'formula': -17.68100578421954, 'futbol': -17.68100578421954)

Por el método de Naive Bayes determinamos que la clasificación para este documento es: 'futbol'

Tiempo de ejecución: 0.017001 segundos

Process finished with exit code 0

*Clasificación: 'futbol'*

*Tiempo de ejecución: 0.017001 segundos*



- “adicionaltenis1”

```

236 def entrenamiento_knn():
237
238
239 def entrenamiento_naive_bayes():
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

*Clasificación: ‘tenis’*

*Tiempo de ejecución: 0.016001 segundos*

- “adicionaltenis2”

```

236 def entrenamiento_knn():
237
238
239 def entrenamiento_naive_bayes():
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

*Clasificación: ‘tenis’*

*Tiempo de ejecución: 0.025002 segundos*

## Conclusiones

- Ambos clasificadores han clasificado correctamente todos los textos adicionales.
- Tanto el proceso de entrenamiento como el de clasificación por Naive Bayes son destacablemente más eficientes que por kNN.

¿Qué necesito para poder ejecutar el programa correctamente?

- Python 3.6.0
- Kit de herramientas de lenguaje natural NLTK

Si no se tiene descargado NLTK, se adjunta un fichero “nltk\_download.py” en el Workspace del proyecto. Para su descarga solo será necesario ejecutar este archivo con su IDE correspondiente.

**Nota:** si al ejecutar el fichero “nltk\_download.py” hubiera algún error, será necesario instalar NLTK previamente siguiendo los pasos de esta fuente:  
“<http://www.nltk.org/install.html>”

¿Cómo añadir una categoría y sus palabras clave?

- 1) Accedo a la ruta del proyecto “.../Workspace/entrenamiento/categorías”
- 2) Creo un archivo “.txt” cuyo título (sin acentos) será el nombre de la categoría que quiero añadir
- 3) Cada palabra clave que quiera añadir se escribirá en una línea distinta del documento

**Nota:** si el nombre de la categoría a añadir contiene un número, éste se debe omitir o determinar un nombre identificativo alternativo sin números para dicha categoría.

¿Cómo añadir documentos al conjunto de entrenamiento?

- 1) Accedo a la ruta del proyecto “.../Workspace/entrenamiento/textos”
- 2) Creo un archivo “.txt” con el formato “{categoría}{idTextoCategoría}.txt”. Ejemplo: para la categoría “futbol” los textos a añadir serán “futbol1.txt”, “futbol2.txt”, etc.
- 3) El interior del archivo debe contener el contenido del documento que queremos añadir como texto de entrenamiento.

### ¿Cómo añado documentos adicionales para probar los clasificadores?

- 1) Accedo a la ruta del proyecto ".../Workspace/adicionales"
- 2) Creo un archivo ".txt" con el formato "*adicional{categoría}{idTextoAdicionalCategoría}.txt*". Ejemplo: para la categoría "futbol" los textos adicionales a añadir serán "adicionalfutbol1.txt", "adicionalfutbol2.txt", etc.
- 3) El interior del archivo debe contener el contenido del documento que queremos añadir como texto adicional.

### ¿Cómo ejecuto los procesos de entrenamiento y clasificación?

- 1) Abro con el IDE correspondiente el archivo "metodos.py" que se encuentra en la ruta del proyecto ".../Workspace"
- 2) Para ejecutar los procesos de entrenamiento nos iremos al final del documento y solo dejaremos descomentada la línea correspondiente al entrenamiento que queremos ejecutar. Ejemplo: `entrenamiento_knn()`

```
378 ▶ if __name__ == "__main__":  
379     entrenamiento_knn()  
380     # entrenamiento_naive_bayes()  
381     # clasificar_documento_knn("adicionaltenis2.txt", 7) # Introducir documento y 'k' deseado  
382     # clasificar_documento_naive_bayes("adicionaltenis2.txt") # Introducir documento  
383  
384  
385  
386  
387
```

- 3) Para ejecutar los procesos de clasificación nos iremos al final del documento y solo dejaremos descomentada la línea correspondiente al clasificador que queremos ejecutar. Ejemplo: `clasificar_documento_knn("adicionaltenis2.txt", 7)`

```
377  
378 ▶ if __name__ == "__main__":  
379     # entrenamiento_knn()  
380     # entrenamiento_naive_bayes()  
381     clasificar_documento_knn("adicionaltenis2.txt", 7) # Introducir documento y 'k' deseado  
382     # clasificar_documento_naive_bayes("adicionaltenis2.txt") # Introducir documento  
383  
384  
385  
386  
387
```

**Nota:** los resultados de la clasificación se mostrarán por consola.

➤ TEXTOS DE ENTRENAMIENTO

tenis1: <http://www.marca.com/tenis/2017/11/26/5a1a9e10ca4741f75c8b458b.html>

tenis2: <http://www.marca.com/tenis/copa-davis/2017/11/25/5a19975622601d12168b457c.html>

tenis3: <http://www.marca.com/tenis/2017/11/21/5a14711e22601d9e058b4649.html>

tenis4: <http://www.marca.com/paralimpicos/2017/11/21/5a143d7322601da7058b4616.html>

tenis5: <http://www.marca.com/tenis/2017/11/20/5a12a350468aeb77118b45f2.html>

tenis6: <http://www.marca.com/tenis/copa-masters/2017/11/19/5a11ed9be5fdea4e378b45c3.html>

tenis7: <http://www.marca.com/tenis/copa-masters/2017/11/18/5a10b03fe2704e04288b463a.html>

tenis8: <http://www.marca.com/tenis/copa-masters/2017/11/18/5a107f4de2704ec76e8b4636.html>

tenis9: <http://www.marca.com/tenis/copa-masters/2017/11/18/5a105892e2704e12218b461c.html>

tenis10: <http://www.marca.com/tenis/copa-masters/2017/11/17/5a0f5165ca4741cc118b4595.html>

tenis11: <http://www.marca.com/tenis/copa-masters/2017/11/17/5a0eff11ca47414a588b467e.html>

tenis12: <http://www.marca.com/tenis/us-open/2017/09/11/59b5ca3ce5fdea245f8b4647.html>

tenis13: <http://www.marca.com/tenis/us-open/2017/09/09/59b38f6dca47418d6f8b4584.html>

tenis14: <http://www.marca.com/tenis/us-open/2017/09/09/59b3256a268e3e15718b45cb.html>

tenis15: <http://www.marca.com/tenis/us-open/2017/09/08/59b2d758e2704e7a3a8b45b8.html>

formula1: <http://www.marca.com/motor/formula1/gp-abu-dhabi/2017/11/27/5a1c1fc6268e3e75498b4694.html>

formula2: <http://www.marca.com/motor/formula1/gp-abu-dhabi/2017/11/27/5a1c1176268e3e737c8b466d.html>

formula3: <http://www.marca.com/motor/formula1/gp-abu-dhabi/2017/11/27/5a1c099a268e3ec7718b464c.html>

formula4: <http://www.marca.com/motor/formula1/gp-abu-dhabi/2017/11/27/5a1c0083268e3e89048b4616.html>

formula5: <http://www.marca.com/motor/formula1/gp-abu-dhabi/2017/11/26/5a1ab94922601dfd148b45f9.html>

formula6: <http://www.marca.com/motor/formula1/gp-abu-dhabi/2017/11/25/5a1997bee5fdea080e8b467c.html>

formula7: <http://www.marca.com/motor/formula1/gp-abu-dhabi/2017/11/25/5a196747468aeb12058b4607.html>

formula8: <http://www.marca.com/motor/formula1/gp-abu-dhabi/2017/11/25/5a196386e5fdea72308b4662.html>

formula9: <http://www.marca.com/motor/formula1/gp-abu-dhabi/2017/11/25/5a1944a5e2704e945d8b45c8.html>

formula10: <http://www.marca.com/motor/formula1/gp-abu-dhabi/2017/11/24/5a184650468aeb5f038b45ab.html>

formula11: <http://www.marca.com/motor/formula1/gp-abu-dhabi/2017/11/24/5a17fd0146163fd4488b4628.html>

formula12: <http://www.marca.com/motor/formula1/gp-abu-dhabi/2017/11/24/5a17db0546163f62548b460f.html>

formula13: <http://www.marca.com/motor/formula1/gp-abu-dhabi/2017/11/23/5a173208e2704e3e5a8b45ea.html>

formula14:

<http://www.marca.com/motor/formula1/2017/11/22/5a15aa3e268e3e133a8b45ac.html>

formula15:

<http://www.marca.com/motor/formula1/2017/11/22/5a153380e5fdea73208b456d.html>

ciclismo1: <http://www.marca.com/ciclismo/giro-italia/2017/11/26/5a1ae94c22601df20e8b4632.html>

ciclismo2:

<http://www.marca.com/ciclismo/2017/11/24/5a16fa6922601d6e498b4575.html>

ciclismo3:

<http://www.marca.com/ciclismo/2017/11/22/5a15e0e8e5fdea7f0a8b4620.html>

ciclismo4:

<http://www.marca.com/ciclismo/2017/11/20/5a0ed80aca4741d0588b46ac.html>

ciclismo5:

<http://www.marca.com/ciclismo/2017/11/27/5a1bfaf6268e3e9f3e8b45f6.html>

ciclismo6:

<http://www.marca.com/ciclismo/2017/11/27/5a1bf88a268e3e891b8b4604.html>

ciclismo7:

<http://www.marca.com/ciclismo/2017/11/18/5a10855122601d2e1f8b4655.html>

ciclismo8:

<http://www.marca.com/ciclismo/2017/11/17/5a0ece09ca474129438b466e.html>

ciclismo9:

<http://www.marca.com/ciclismo/2017/11/15/5a0c2c18268e3ed21f8b45a2.html>

ciclismo10:

<http://www.marca.com/ciclismo/2017/11/14/5a0ad2c9e2704eee518b4630.html>

ciclismo11:

<http://www.marca.com/ciclismo/2017/11/13/5a08bacf468aeb1248b45cb.html>

ciclismo12:

<http://www.marca.com/ciclismo/mundial/2017/09/24/59c3eddbca474109068b46c5.html>

ciclismo13:

<http://www.marca.com/ciclismo/mundial/2017/09/24/59c7c50d468aebdd108b461f.html>

ciclismo14:

<http://www.marca.com/ciclismo/mundial/2017/09/24/59c7f600468aebc2648b4671.html>

ciclismo15:

<http://www.marca.com/ciclismo/mundial/2017/09/23/59c657c5e5fdea3e6e8b4574.html>

baloncesto1:

<http://www.marca.com/baloncesto/nba/2017/11/26/5a1a705fca4741ac6c8b4578.html>

baloncesto2:

<http://www.marca.com/baloncesto/nba/2017/11/26/5a1a78e7ca474114348b457e.html>

baloncesto3:

<http://www.marca.com/baloncesto/nba/2017/11/26/5a1a7552ca47412f6a8b4574.html>

baloncesto4:

<http://www.marca.com/baloncesto/nba/2017/11/26/5a1a7b6bca4741684d8b457b.html>

baloncesto5:

<http://www.marca.com/baloncesto/nba/2017/11/26/5a1a8a1bca474125568b457f.html>

baloncesto6:

<http://www.marca.com/baloncesto/nba/2017/11/26/5a1a85c1ca474193528b457a.html>

baloncesto7:

<http://www.marca.com/baloncesto/nba/2017/11/25/5a19339a268e3e5c1d8b459c.html>

baloncesto8:

<http://www.marca.com/baloncesto/nba/2017/11/25/5a192704268e3e641c8b4599.html> (1ra parte)

baloncesto9:

<http://www.marca.com/baloncesto/nba/2017/11/25/5a192704268e3e641c8b4599.html> (2da parte)

baloncesto10:

<http://www.marca.com/baloncesto/nba/2017/11/25/5a192704268e3e641c8b4599.html> (3ra parte)

baloncesto11:

<http://www.marca.com/baloncesto/nba/2017/11/26/5a1a7f6dca474118658b4584.html>

baloncesto12:

<http://www.marca.com/baloncesto/nba/2017/11/25/5a1918c1268e3ea0088b4596.html>

baloncesto13:

<http://www.marca.com/baloncesto/nba/2017/11/25/5a191160268e3e5a278b458a.html>

baloncesto14:

<http://www.marca.com/baloncesto/nba/2017/11/25/5a190dbc268e3ecd768b459b.html>

baloncesto15:

<http://www.marca.com/baloncesto/nba/2017/11/23/5a16bbaa468aeb8258b45fb.html>

futbol1: [http://www.marca.com/eventos/marcador/futbol/2017\\_18/liga/jornada\\_10/ala\\_val/](http://www.marca.com/eventos/marcador/futbol/2017_18/liga/jornada_10/ala_val/)

futbol2: [http://www.marca.com/eventos/marcador/futbol/2017\\_18/liga/jornada\\_12/pal\\_lev/](http://www.marca.com/eventos/marcador/futbol/2017_18/liga/jornada_12/pal_lev/)

futbol3:

<http://www.marca.com/futbol/atletico/2017/11/26/5a19ee50e5fdeaeb648b4573.html>

futbol4: <http://www.marca.com/claro-mx/futbol/mexicanos-mundo/2017/11/25/5a19c4b1ca4741bd4b8b463d.html>

futbol5:

<http://www.marca.com/futbol/girona/2017/11/25/5a19a8b1468aeb36028b4620.html>

futbol6: [http://www.marca.com/eventos/marcador/futbol/2017\\_18/liga/jornada\\_13/rma\\_mga/](http://www.marca.com/eventos/marcador/futbol/2017_18/liga/jornada_13/rma_mga/)

futbol7: [http://www.marca.com/eventos/marcador/futbol/2017\\_18/liga/jornada\\_13/ala\\_eib/](http://www.marca.com/eventos/marcador/futbol/2017_18/liga/jornada_13/ala_eib/)

futbol8: [http://www.marca.com/eventos/marcador/futbol/2017\\_18/liga/jornada\\_13/cel\\_leg/](http://www.marca.com/eventos/marcador/futbol/2017_18/liga/jornada_13/cel_leg/)

futbol9:

<http://www.marca.com/futbol/celta/2017/11/25/5a18af67e5fdea221f8b4619.html>

futbol10: [http://www.marca.com/eventos/marcador/futbol/2017\\_18/liga/jornada\\_10/get\\_rso/](http://www.marca.com/eventos/marcador/futbol/2017_18/liga/jornada_10/get_rso/)

futbol11:

[http://www.marca.com/eventos/marcador/futbol/2017\\_18/champions/1a\\_fase/jornada\\_5/grupo\\_e/sev\\_liv/](http://www.marca.com/eventos/marcador/futbol/2017_18/champions/1a_fase/jornada_5/grupo_e/sev_liv/)

futbol12: [http://www.marca.com/eventos/marcador/futbol/2017\\_18/liga/jornada\\_12/sev\\_cel/](http://www.marca.com/eventos/marcador/futbol/2017_18/liga/jornada_12/sev_cel/)

futbol13: [http://www.marca.com/eventos/marcador/futbol/2017\\_18/liga/jornada\\_12/ath\\_vil/](http://www.marca.com/eventos/marcador/futbol/2017_18/liga/jornada_12/ath_vil/)



futbol14:

[http://www.marca.com/eventos/marcador/futbol/2017\\_18/europa\\_league/1a\\_fase/jornada\\_5/grupo\\_a/asn\\_vil/](http://www.marca.com/eventos/marcador/futbol/2017_18/europa_league/1a_fase/jornada_5/grupo_a/asn_vil/)

futbol15: [http://www.marca.com/eventos/marcador/futbol/2017\\_18/la-liga/jornada\\_11/ala\\_esp/](http://www.marca.com/eventos/marcador/futbol/2017_18/la-liga/jornada_11/ala_esp/)

futbol16: [http://www.marca.com/eventos/marcador/futbol/2017\\_18/la-liga/jornada\\_12/mga\\_dep/](http://www.marca.com/eventos/marcador/futbol/2017_18/la-liga/jornada_12/mga_dep/)

futbol17: [http://www.marca.com/eventos/marcador/futbol/2017\\_18/la-liga/jornada\\_10/mga\\_cel/](http://www.marca.com/eventos/marcador/futbol/2017_18/la-liga/jornada_10/mga_cel/)

futbol18:

[http://www.marca.com/eventos/marcador/futbol/2017\\_18/copa/dieciseisavos/ida/num\\_mga/](http://www.marca.com/eventos/marcador/futbol/2017_18/copa/dieciseisavos/ida/num_mga/)

futbol19: [http://www.marca.com/eventos/marcador/futbol/2017\\_18/la-liga/jornada\\_7/sev\\_mga/](http://www.marca.com/eventos/marcador/futbol/2017_18/la-liga/jornada_7/sev_mga/)

futbol20:

[http://www.marca.com/eventos/marcador/futbol/2017\\_18/europa\\_league/1a\\_fase/jornada\\_5/grupo\\_l/ros\\_rso/](http://www.marca.com/eventos/marcador/futbol/2017_18/europa_league/1a_fase/jornada_5/grupo_l/ros_rso/)

➤ TEXTOS ADICIONALES

adicionalfutbol1: <http://www.marca.com/futbol/real-madrid/2017/11/19/5a11ea9622601d8f608b45ae.html>

adicionalfutbol2:  
<http://www.marca.com/futbol/betis/2017/11/28/5a1d2f1b22601deb668b45eb.html>

adicionaltenis1:  
<http://www.marca.com/tenis/2017/07/16/596b9f66e5fdea324e8b45e9.html>

adicionaltenis2:  
<http://www.marca.com/tenis/wimbledon/2017/07/16/596b7d8ce2704ef9188b4663.html>

adicionalformula1: <http://www.marca.com/motor/formula1/gp-abu-dhabi/2017/11/28/5a1d2e2c268e3e4f688b45b6.html>

adicionalformula2:  
<http://www.marca.com/motor/formula1/2017/11/27/5a1c3a2946163f412b8b45e9.html>

adicionalciclismo1:  
<http://www.marca.com/ciclismo/2017/11/28/5a1cc1db22601dca6e8b461a.html>

adicionalciclismo2:  
<http://www.marca.com/ciclismo/2017/11/28/5a1cc68022601ded658b4628.html>

adicionalbaloncesto1:  
<http://www.marca.com/baloncesto/euroliga/2017/11/28/5a1c6fa3468aeb4c5f8b4653.html>

adicionalbaloncesto2:  
<http://www.marca.com/baloncesto/nba/2017/11/28/5a1d067422601df0658b4620.html>