Prueba Técnica Profesional III Departamento de Datos no Estructurados: Prueba Imágenes con OCR.

Manuel Alejandro Diaz Rubiano.

Cel: 3196371560

Email: alejandromadr@gmail.com

DAVIVIENDA

Entendimiento del Negocio.

Según Gartner, en este momento, los datos no estructurados conforman el 80-90% del total de los datos que manejan las compañías. Dentro de ellos, las imágenes de documentos ocupan un lugar importante. Cada año se hace más evidente la necesidad de convertir los datos contenidos en estas imágenes en información útil que pueda ser analizada en bases de datos.

Dentro de los primeros pasos para lograr este objetivo se encuentra, por supuesto, el identificar qué imágenes guardan contenido y cuáles pueden desecharse, lo que se traducirá en menores costos de almacenamiento, procesamiento de archivos y recursos humanos.

Objetivo.

El objetivo de este reto es lograr un filtro que discrimine automáticamente un tipo de documento

sin información relevante: páginas en blanco. Se busca que este filtro reciba como entrada una

carpeta con imágenes de documentos diversos y produzca como salida dos carpetas, una con

imágenes de páginas sin contenido y otra con imágenes de páginas con contenido.

Con contenido

Sin contenido

Sin contenido

Sin contenido

Exploración de datos.

Se tienen dos conjuntos de bases de datos, con archivos escaneados. La primera carpera se llama “blanco”, y contiene documentos escaneados sin nada de texto, o con texto para nada relevante, como marcas de agua o pie de páginas, la otra carpeta se llama “documentos” y contiene información relevante, texto, datos, y demás.

Ejemplo de los documentos en Blanco.

page-156

page-157

page-158

page-159

page-160

page-161

page-162

page-163

Ejemplo de los documentos con Contenido.

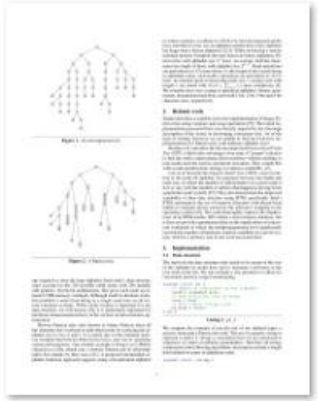
page-2

page-3

page-4

page-6

page-7



page-8

page-9

page-10

Herramienta principal para la lectura de archivos.

- El procesamiento de datos se realiza en el lenguaje de programación Python, y se puede observar ejemplos de como lee los archivos a continuación:

A DETAILED ANALYSIS OF THE
COMPONENT OBJECT MODEL

Thesis approved:

Dr. M. H. Samadzadeh
Thesis Advisor

Dr. Blayne E. Mayfield

Dr. Nohpill Park

Dr. Gordon Emslie
Dean of the Graduate College

img1:
A DETAILED ANALYSIS OF THE

COMPONENT OBJECT MODEL

Thesis approved:

Dr. M. H. Samadzadeh

Thesis Advisor

Dr. Blayne E. Mayfield

Dr. Nohpill Park

Dr. Gordon Emslie

Dean of the Graduate College

DISCLAIMER

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

AMD's products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD's product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

Advanced Micro Devices, Inc.
One AMD Place
P.O. Box 3453
Sunnyvale, CA 94088-3453
www.amd.com

For AMD Accelerated Parallel Processing:

URL:	developer.amd.com/appsdk
Developing:	developer.amd.com/
Forum:	developer.amd.com/openclforum

img2:
DISCLAIMER

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

AMD's products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications

Intended to support or sustain life, or in any other application in which the failure of AMD's product could create a situation where personal injury, death, or severe property or envi-

ronmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

AMD‡1

Advanced Micro Devices, Inc.
One AMD Place
P.O. Box 3453
Sunnyvale, CA 94088-3453
www.amd.com

For AMD Accelerated Parallel Processing:

URL: developer.amd.com/appsdk
Developing: developer.amd.com/
Forum: developer.amd.com/openclforum

URL:
Developing:
Forum:

Se puede observar en los dos slides anteriores, que el programa logra interpretar bastante bien el texto de los documentos escaneados.

Las herramientas implementadas fue la librería de código abierto OpenCV, y la herramienta Tesseract, que es un motor de reconocimiento óptico de caracteres para varios sistemas operativos, desarrollado por Google.

La técnica utilizada para el reconocimiento de caracteres, fue la psm-3, que realiza segmentación de páginas totalmente automática, pero sin orientación ni detección de guiones.

Técnicas y Herramientas Utilizadas.

Dificultades encontradas.

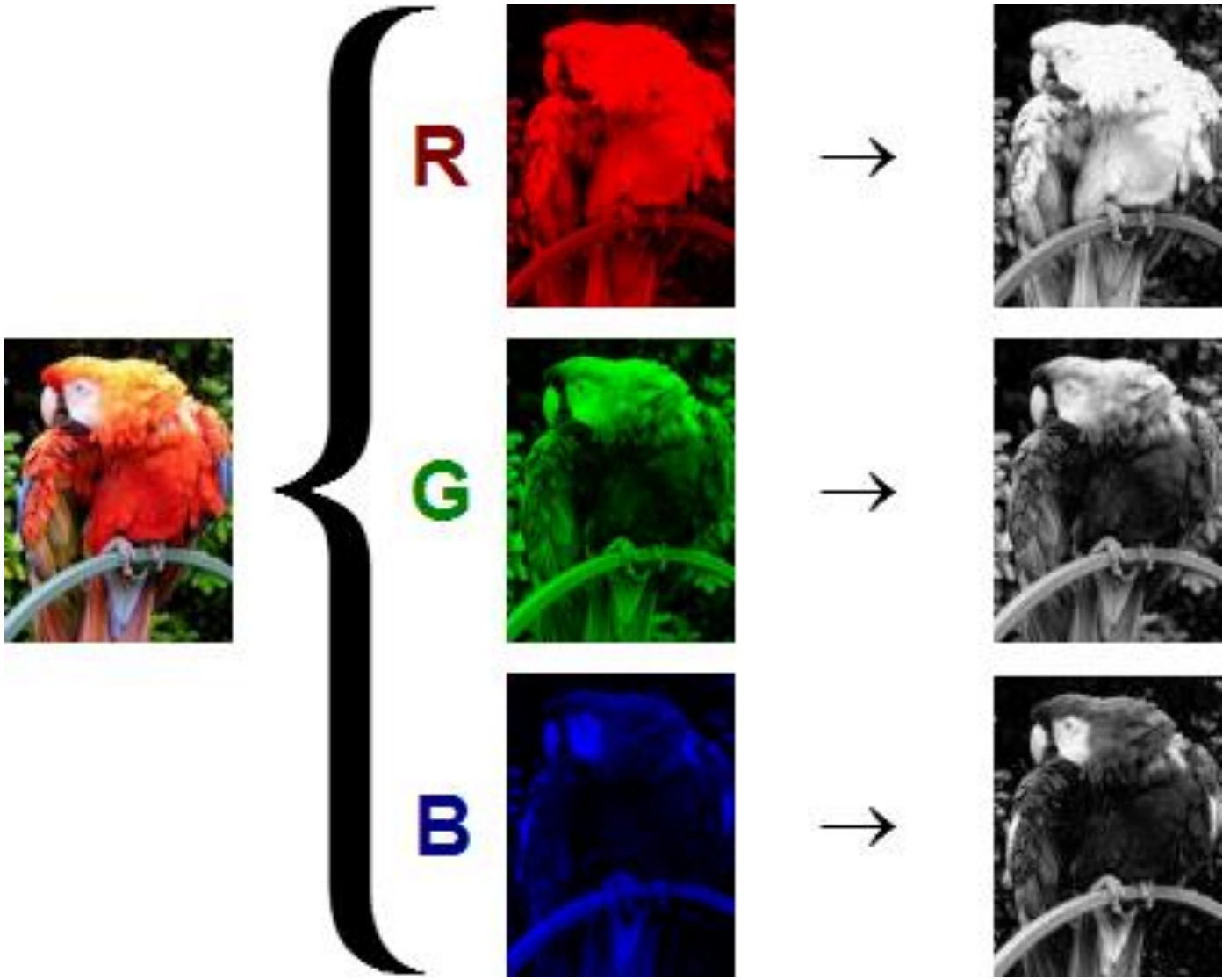
- Algunas de las dificultades que se encontraron, fue que el programa detectaba incluso la letra mas pequeña, o mas borrosa, y lo clasificaba como contenido.
- Las marcas de agua, o los pie de pagina, eran detectados por el programa, lo que lo hacia clasificar como un documento con contenido.
- Ya tenemos la forma de como reconocer y extraer los textos de las imágenes, pero se debía encontrar la manera de clasificar los documentos entre con contenido, y en blanco.

Aspectos importantes a tener en cuenta.

- Algunos documentos tienen imágenes.
- Algunas imágenes son a color.
- Los archivos en blanco, tienen algunos caracteres muy vagos pero reconocibles.
- Los documentos no son 100% blanco y negro.

Planteamiento y solución.

Todas las imágenes tienen un espectro colores en la escala RGB, conformada por tres matrices con valores numéricos, y dependiendo de los valores numéricos en esas tres matrices, es que se configura la forma de la imagen, y principalmente, se configura la coloración; se pueden extraer de ahí la cantidad de pixeles por los cuales está conformada una imagen.



Original

Modified

- Estas imágenes se pueden pasar a una escala grises, en la cual la imagen quedara a blanco y negro.
- Al momento de pasar la imagen a escala de grises, se pasa de tener 3 matrices en la escala RGB a 2 matrices, en la cual se configura una matriz para la intensidad del color blanco, y otra para la intensidad del color negro.
- Gracias a la ayuda de la librería numérica de Python “Numpy”, se puede contar la cantidad de pixeles de color oscuro, y la cantidad de pixeles claros.

Journal of Economic Literature,
Vol. XXXIX (December 2001) pp. 1137–1176

The Determinants of Earnings:
A Behavioral Approach

SAMUEL BOWLES, HERBERT GINTIS,
and MELISSA OSBORNE[1]

race and sex in the United States, between two-thirds and four-fifths of the variance of the natural logarithm of hourly wages or of annual earnings is unexplained by the above variables. Some of the unexplained variance is contributed by the transitory component of earnings and response error (Gary Solon 1992; David Zimmerman 1992; Bowles 1972). But this leaves well

Durlauf, Paula England, Henry Farber, Daniel Hamermesh, Karla Hoff, Min-Hsiung Huang, Melvin Kohn, Michael Kremer, Alan Krueger, Charles Manski, Casey Mulligan, Richard Murnane, William Nordhaus, Mark Rosenzweig, Cecilia Rouse, and Eric Olin Wright for providing unpublished estimates, comments and other assistance, as well as participants at seminars at Yale University, University of Chicago, MIT, and University of Wisconsin, and the *Journal of Economic Literature*'s anonymous reviewers, for comments. Bridget Longridge for research assistance, and the MacArthur Foundation for financial support.

1137

¿Cómo puede ayudar esto?

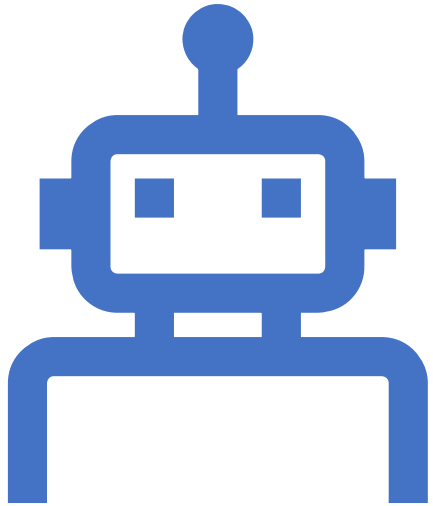
Si los documentos que se escanean contienen texto principal, que no sean marcas de agua ni demás, al momento de pasar la imagen a escala de grises, ese texto serán pixeles con valores netamente negros, o con el valor 0, y los espacios en blanco serán representados por pixeles blancos, o con el valor 255. La hipótesis para entrenar el modelo es que los documentos escaneados que no tienen contenido, o que están en blanco, tienen menos pixeles, principalmente menos pixeles negros, a comparación de los documentos que tienen contenido. Al tener el recuento de la cantidad de pixeles, se puede entrenar un modelo clasificatorio, para que aprenda la diferencia entre documentos en blanco y documentos con contenido, gracias a la contabilización de los pixeles en las imágenes.

Modelamiento.

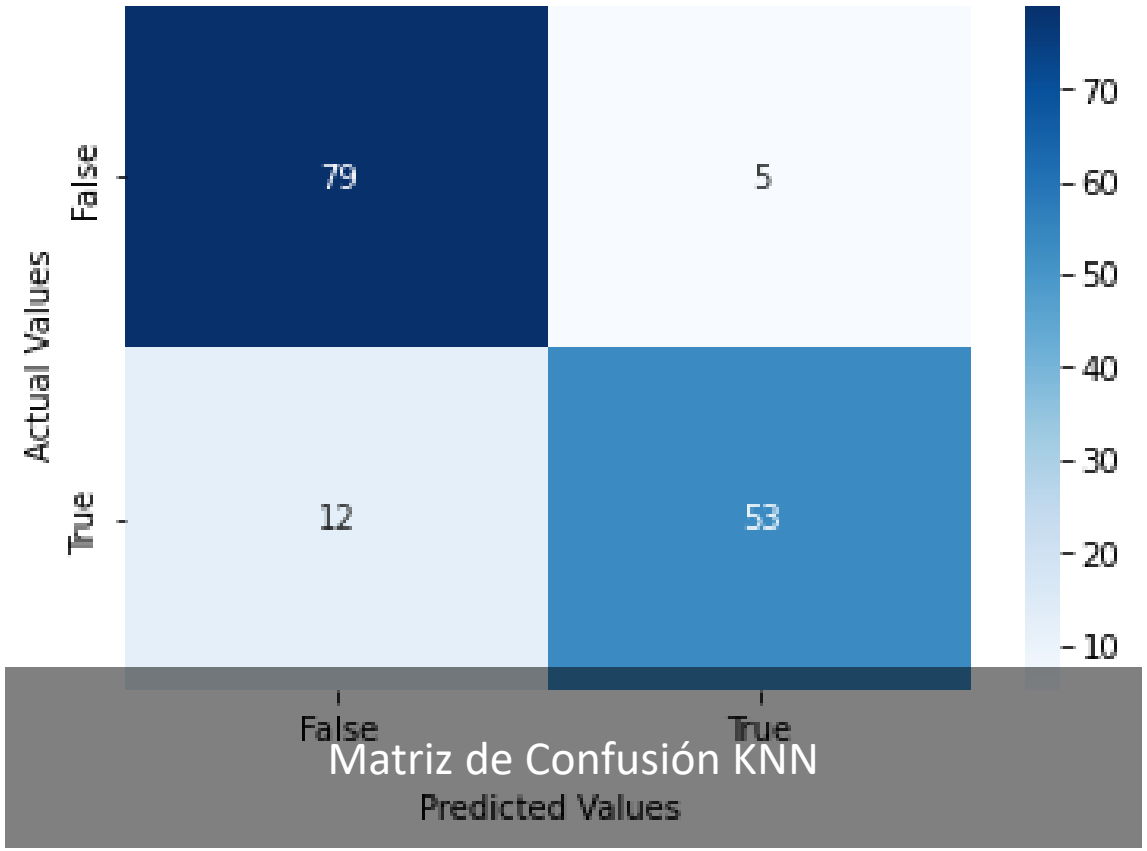
- Sera un problema de aprendizaje supervisado.
- Se creara una variable binaria, con el nombre texto, y será 0, si el documento esta clasificado como un documento en blanco, y 1 si el documento esta clasificado como un documento con contenido.
- Se divide la base en entrenamiento y prueba, para poder verificar el rendimiento del algoritmo.
- Se utiliza el 60% de la base para entrenamiento, y el 40% restante para testing.

Algoritmos testeados.

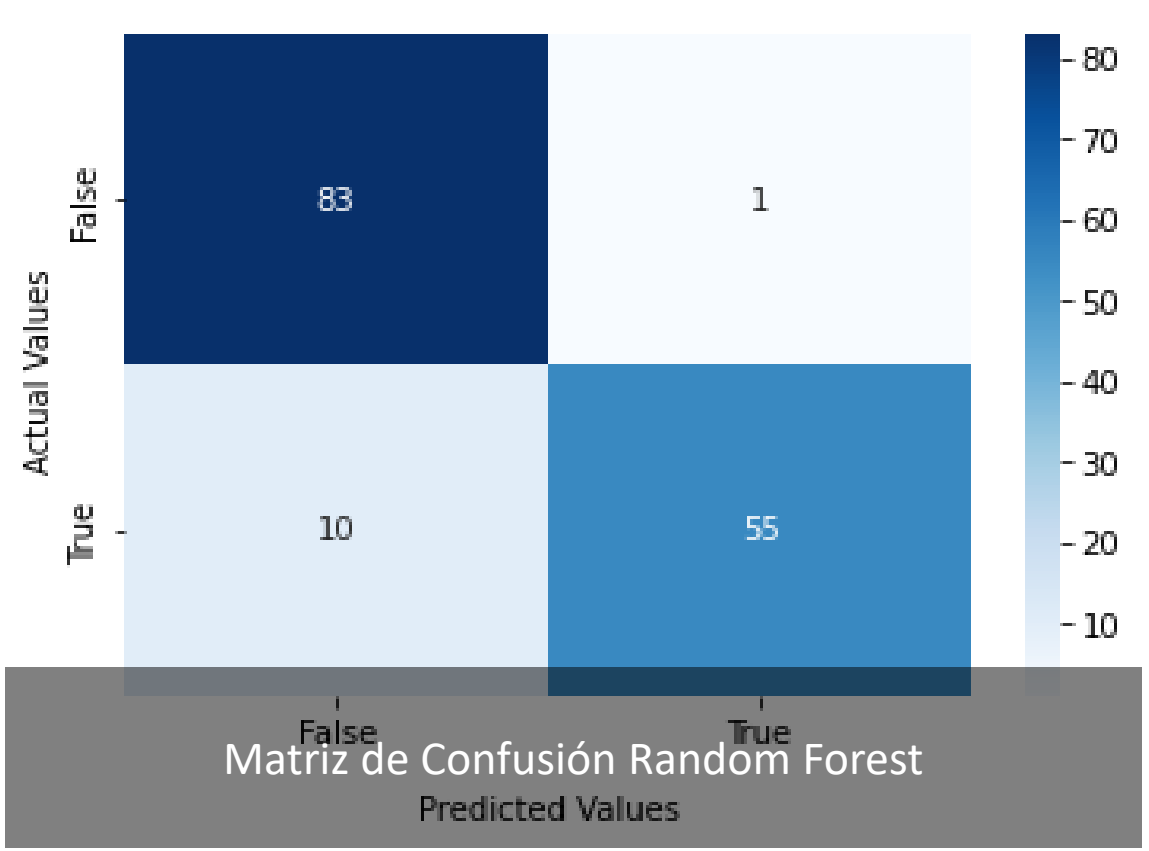
- K vecinos mas cercanos.
- Naive Bayes.
- Regresión Logística.
- Clasificador Random Forest.



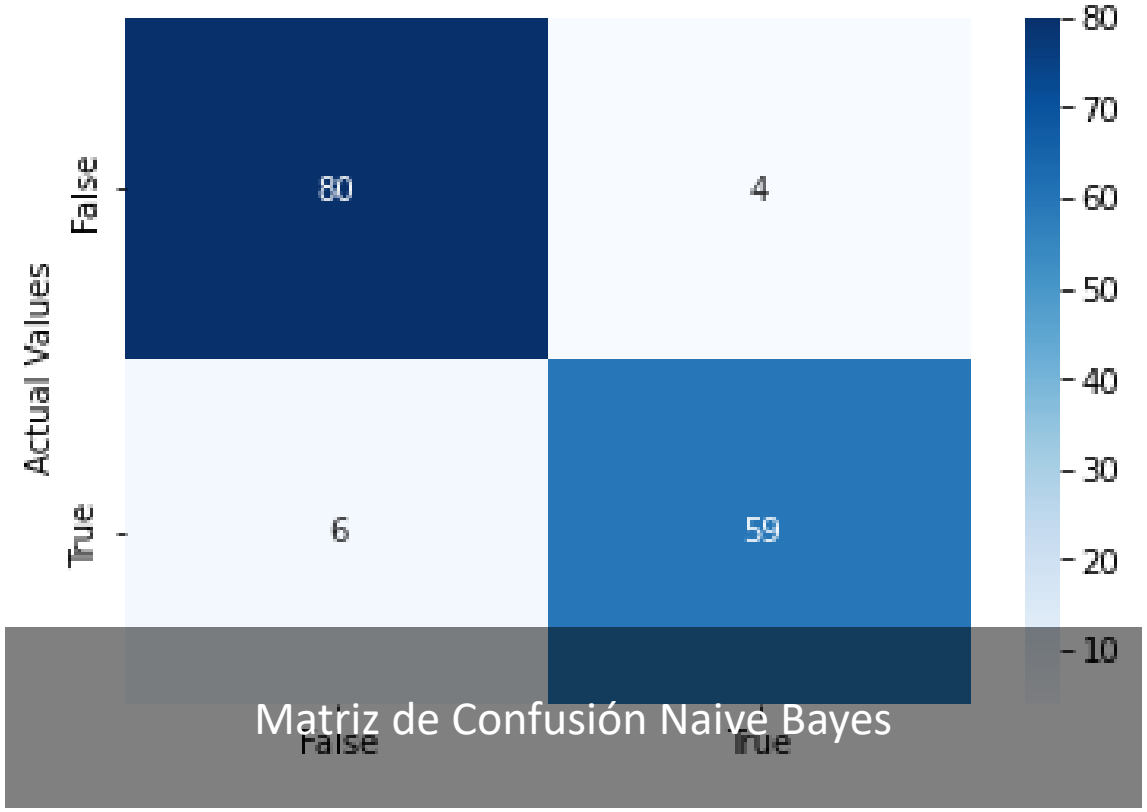
Resultados y Evaluación.



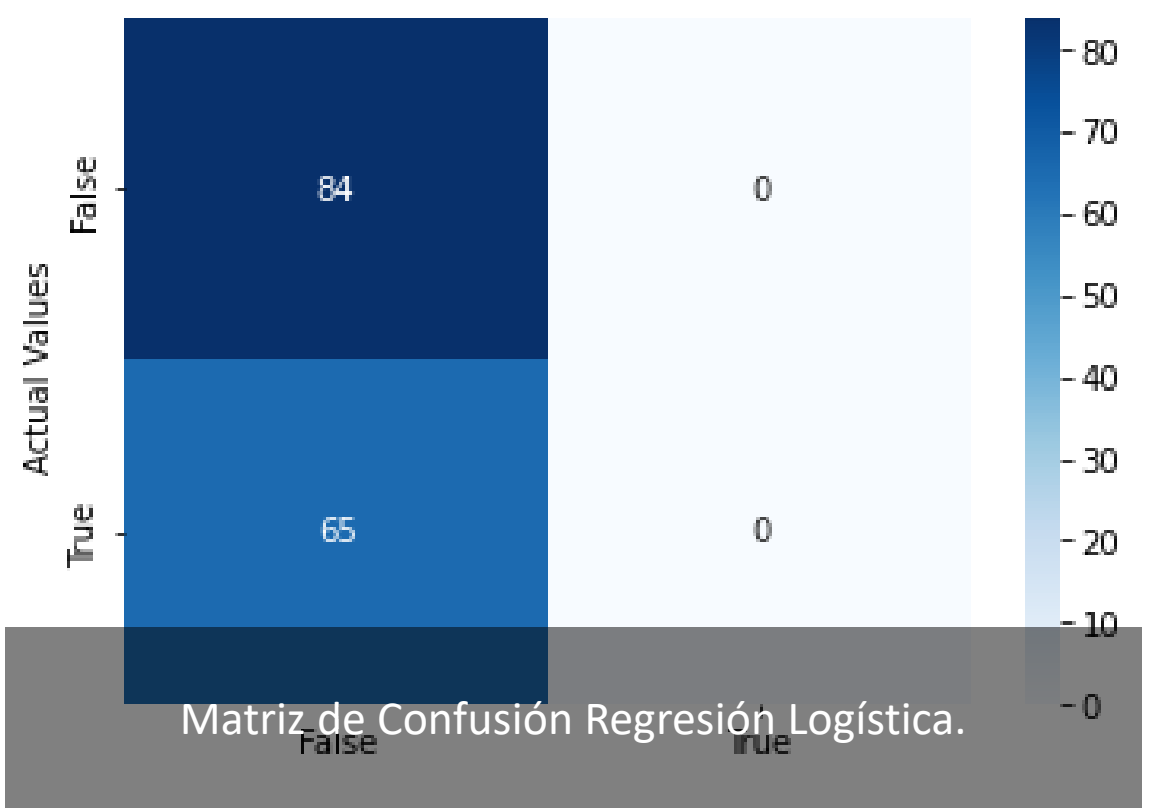
Matriz de Confusión KNN



Matriz de Confusión Random Forest



Matriz de Confusión Naive Bayes



Matriz de Confusión Regresión Logística.

Evaluación de Métricas.

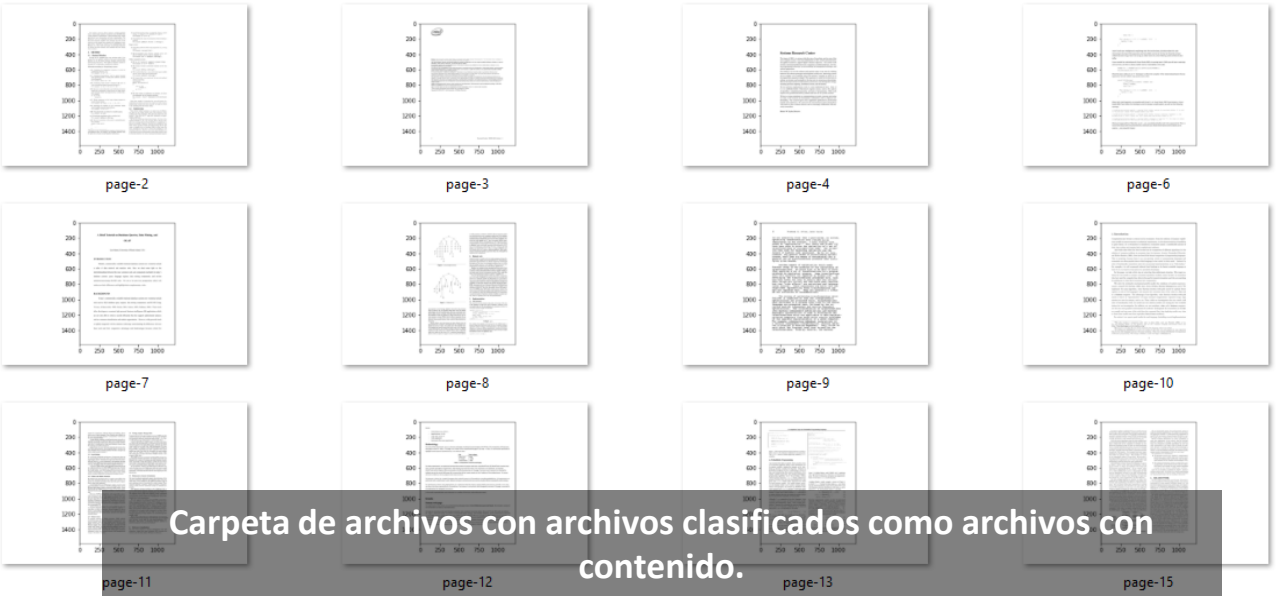
Modelo	Accuracy	Precision	Recall
Regresion Logistica	0.5637	0.0	0.0
Naive Bayes	0.932885	0.936507	0.907692
KNN	0.885906	0.913793	0.815384
Random Forest	0.926174	0.982142	0.846153

Las métricas escogidas para evaluar los modelos, son la matriz de confusión, el accuracy, precision y recall.

Clasificar imágenes y cargarlas a la carpeta correspondiente.

Se escogió como mejor modelo, el modelo Naive Bayes.

Con este modelo, se clasifica para todos los archivos, si tienen contenido o no tienen contenido. Posteriormente, se crear un ciclo, en el cual se verifica su clasificación, y si tiene contenido, la imagen (que esta en forma matricial almacenada en un diccionario de Python), se envía a la carpeta docs_con_contenido, y si esta clasificada que no tiene contenido, se envía a la carpeta docs_sin_contenido.



Carpeta de archivos con archivos clasificados como archivos con contenido.

Carpeta de archivos con archivos clasificados como archivos sin contenido.

Ejemplo de las carpetas.

Conclusiones.

- Se puede observar los mejores modelos, de los cuales se destacan dos, que son Random Forest y Naive Bayes.
- Estos algoritmos son fáciles de interpretar, y su costo de computación es bajo.
- El resultado del *accuracy*, con un 0,93 para para Naive Bayes, y 0,92 para Random Forest.
- Se pueden ahora clasificar los documentos escaneados, entre si están en blanco, o si tienen contenido.