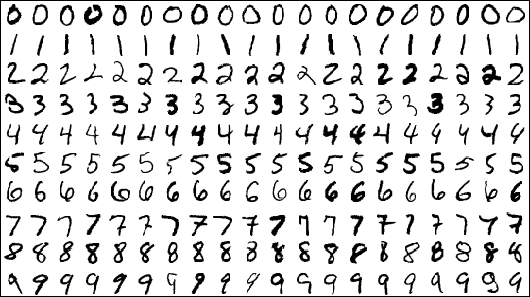
**Reconocimiento de imágenes usando el paquete MNIST**

La base de datos MNIST (Modified National Institute of Standars and Technology) es una gran base de datos de dígitos escritos a mano que se usa comúnmente para capacitar a varios sistemas de procesamiento de imágenes, cada imagen es un cuadrado de 28 x 28 píxeles con valores de escala de grises. La base MNIST contiene 60.000 imágenes destinadas al entrenamiento del modelo y 10.000 para validación. A medida que surgen nuevas técnicas de aprendizaje automático, MNIST sigue siendo un recurso confiable para investigadores y estudiantes.



**Figura 1.** Dígitos manuscritos.

**Objetivos:**

* En este trabajo se abordará el reto de construir clasificadores multiclase utilizando variadas técnicas, entre las cuales se encuentran:

1. Regresión logística multinomial
2. Árboles de clasificación
3. Bosques aleatorios
4. Maquinas de soporte vectorial
5. Redes neuronales

Los métodos anteriormente mencionados se construyen utilizando el conjunto de datos MNIST.

* Validar y comparar los modelos con el fin de identificar el mejor modelo para clasificar los dígitos.

**Definiciones:**

Algunas definiciones básicas que ayudan a la interpretación de los modelos:

**Variable dependiente o respuesta:** Es aquella variable cuyo resultado depende de los valores que tomen otras variables.

**Variable independiente o explicativas:** Es aquella variable cuyo resultado no depende de los valores que tomen otras variables.

**Matriz de confusión:** Es una herramienta que permite la visualización del desempeño de un algoritmo. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Uno de los beneficios de las matrices de confusión es que facilitan ver si el sistema está confundiendo dos clases.

|  |  |  |  |
| --- | --- | --- | --- |
| Observación |  | Predicción | |
| Positivos | Negativos |
| Positivos | Verdaderos positivos | Falsos Negativos |
| Negativos | Falsos Positivos | Verdaderos Negativos |

**Tabla 1.** Matriz de confusión.

Donde:

**Verdaderos positivos:** Cantidad de positivos que fueron clasificados correctamente como positivos por el modelo.

**Falsos Negativos:** cantidad de positivos que fueron clasificados incorrectamente como negativos

**Falsos Positivos:** Cantidad de negativos que fueron clasificados incorrectamente como positivos.

**Verdaderos Negativos**: Cantidad de negativos que se clasifican correctamente como negativos por el modelo.

**Regresión logística multinomial.**

Se procede con el desarrollo del primer objetivo, se comienza utilizando la Regresión logística multinomial, cuya principal característica es que la variable dependiente o respuesta puede tener más de dos categorías, esta variable respuesta puede ser tanto nominal como ordinal, cuantitativa o cualitativa. En este caso, la variable es nominal y cuantitativa.

*Variable dependiente:* Etiqueta (en este caso se tienen 10 etiquetas correspondientes a los números del 0 al 9). A continuación, se presenta el conteo para cada etiqueta:

0 1 2 3 4 5 6 7 8 9

5923 6742 5958 6131 5842 5420 5918 6265 5851 5949

*Variables independientes:* son los 784 pixeles variables de las imágenes (0-255).

Se modela la regresión con el conjunto de datos de MNIST en el software estadístico R mediante el paquete *nnet*, los resultados se presentan a continuación mediante la matriz de confusión:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **0** | 5540 | 0 | 20 | 26 | 4 | 59 | 12 | 42 | 18 | 20 |
| **1** | 10 | 6602 | 80 | 61 | 72 | 69 | 33 | 117 | 359 | 50 |
| **2** | 25 | 34 | 5363 | 200 | 16 | 32 | 14 | 71 | 76 | 14 |
| **3** | 9 | 19 | 64 | 5370 | 6 | 170 | 0 | 12 | 77 | 68 |
| **4** | 28 | 7 | 66 | 20 | 5487 | 69 | 26 | 86 | 61 | 278 |
| **5** | 143 | 10 | 38 | 96 | 11 | 4095 | 63 | 11 | 100 | 23 |
| **6** | 111 | 10 | 101 | 44 | 83 | 214 | 5735 | 4 | 74 | 14 |
| **7** | 7 | 8 | 56 | 60 | 7 | 25 | 4 | 5658 | 23 | 124 |
| **8** | 44 | 42 | 148 | 197 | 24 | 592 | 30 | 25 | 4966 | 80 |
| **9** | 6 | 10 | 22 | 57 | 132 | 95 | 1 | 239 | 97 | 5278 |

**Tabla 2.** Matriz de confusión.

El porcentaje de clasificación correcta es:

Con una probabilidad del 90.1% una observación está clasificada correctamente.

El porcentaje de mala clasificación es:

Hay una probabilidad del 9.8% de que una observación este mal clasificada.

Otras alternativas para medir el ajuste del modelo son usando los R cuadrado-logísticos, es decir, los R- cuadrado de Cox y Snell, R-cuadrado de Nagelkerke, ambos indican la parte de varianza de la variable dependiente explicada por el modelo. Se acostumbra a decir que la parte de la variable dependiente explicada por el modelo oscila entre la R-cuadrado de Cox y Snell y R-cuadrado de Nagelkerke, cuando mas alto sea, mas explicativo es el modelo.

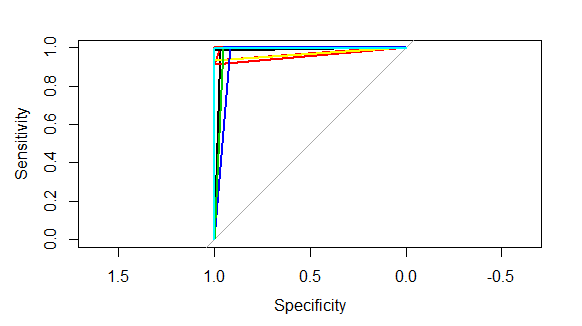
|  |  |
| --- | --- |
| R-cuadrado de Cox y Snell | 0.9777 |
| R-cuadrado de Nagelkerke | 0.9877 |

**Tabla 3.** R-cuadrados.

**Curva ROC:**

Es útil para determinar la bondad de ajuste del modelo, es decir, la probabilidad de que un individuo sea clasificado correctamente. Se entiende que una curva ROC es buena cuando su área bajo la curva toma valores cercanos a 1 y es considerada prueba perfecta, cuando toma el valor de 0.5 se considera una prueba inútil.

|  |  |
| --- | --- |
| Multi-class area under the curve | 0.9956 |



**Figura 2.** Curva ROC

Se observa un area debajo de la curva muy cercana a 1, lo que indica un buen ajuste del modelo.

En general, se evidencia un buen ajuste del modelo ya que la tasa de clasificación correcta de una observación es alta y los R-cuadrados son altos, es decir, explican muy bien el modelo.

**Bosques aleatorios**

El Bagging o Bootstrap es una técnica para reducir varianza de una función de predicción estimada. El Bagging parece funcionar bien para los procedimientos de alta variación y bajo sesgo, como los árboles. Para regresión, simplemente se ajusta el mismo árbol de regresión muchas veces a las versiones muestreadas de los datos de entrenamiento y se promedio el resultado. Para la clasificación, un comité de árboles emite un voto para la clase predicha.

Los bosques aleatorios es una modificación grande del Boosting (Maquinas de gradiente mejorado que trabajan similar al Bagging diferenciándose en que los árboles son construidos secuencialmente. Cada árbol se construye para aprender de los errores de mala clasificación de los anteriores árboles construidos) En muchos problemas, el rendimiento de los bosques aleatorios es muy satisfactorio además de que son fáciles de entrenar. Los bosques aleatorios son muy populares y se implementan en una gran variedad de paquetes.

El modelo se realiza usando la libraría *randomForest* del software estadístico R. Los resultados se presentan a través de la matriz de confusión:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **0** | 5793 | 0 | 18 | 15 | 7 | 24 | 29 | 4 | 27 | 6 |
| **1** | 2 | 6620 | 34 | 18 | 11 | 7 | 8 | 16 | 22 | 4 |
| **2** | 47 | 17 | 5602 | 63 | 40 | 13 | 39 | 57 | 69 | 11 |
| **3** | 16 | 11 | 114 | 5621 | 10 | 134 | 18 | 56 | 97 | 54 |
| **4** | 13 | 15 | 20 | 13 | 5542 | 4 | 35 | 17 | 30 | 153 |
| **5** | 43 | 13 | 25 | 150 | 25 | 5017 | 51 | 5 | 49 | 42 |
| **6** | 47 | 20 | 28 | 4 | 20 | 64 | 5707 | 4 | 23 | 1 |
| **7** | 14 | 24 | 79 | 24 | 53 | 5 | 2 | 5945 | 24 | 95 |
| **8** | 31 | 52 | 77 | 116 | 56 | 94 | 46 | 19 | 5282 | 78 |
| **9** | 24 | 16 | 20 | 77 | 144 | 46 | 9 | 102 | 61 | 5450 |

**Tabla 4.** Matriz de confusión

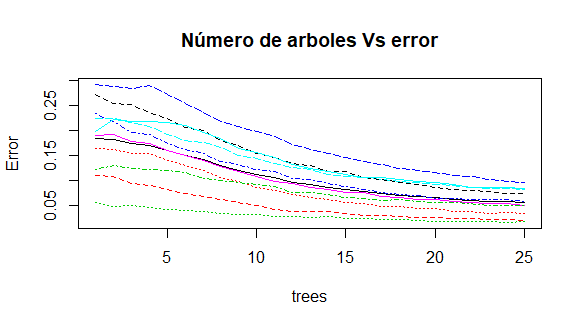
Porcentaje de clasificación correcta:

Con una probabilidad del 86.13% una observación está clasificada correctamente.

Porcentaje de mala clasificación:

Hay una probabilidad del 13.87% de que una observación este mal clasificada.

La tasa de error OOB (observaciones fuera de la bolsa) hace referencia el porcentaje de observaciones que no se usaron para construir los árboles, o también se puede interpretar como una estimación honesta del error de predicción basada en el proceso de Bagging, en este caso, tal porcentaje es de 5.7%.

 **Figura 3.** Grafica de errores vs número de árboles.

En la figura 3, se puede evidenciar que el error disminuye a medida que aumenta el numero de árboles.

De los anteriores resultados se concluye el que ajuste del modelo es bastante bueno, ya que la tasa de buena clasificación es significativamente grande.

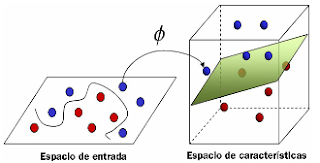
**Máquinas de Soporte Vectorial (SVM)**

Las máquinas de soporte vectorial son una técnica de IA (Inteligencia Artificial) para clasificación; uno de los fundamentos de esta técnica es el uso de algoritmos de vectores soporte lo cual es una generalización no-lineal del algoritmo Semblanza Generalizada desarrollado en Rusia en los años 60's; básicamente son sistemas de aprendizaje que usan un espacio de hipótesis de funciones lineales en un espacio de rasgos de mayor dimensión, entrenadas por un algoritmo proveniente de la teoría de optimización.

Otro tema fundamental es la *minimización del riesgo empírico* lo cual es un algoritmo que se enfoca en clasificar entre miembros positivos y miembros negativos en un problema o situación particular, es decir, el algoritmo busca la superficie optima que clasifica las observaciones, en el caso bidimensional esta superficie será una línea recta, y en el caso de 3 dimensiones será un plano.

**¿Cómo lo hacen?**

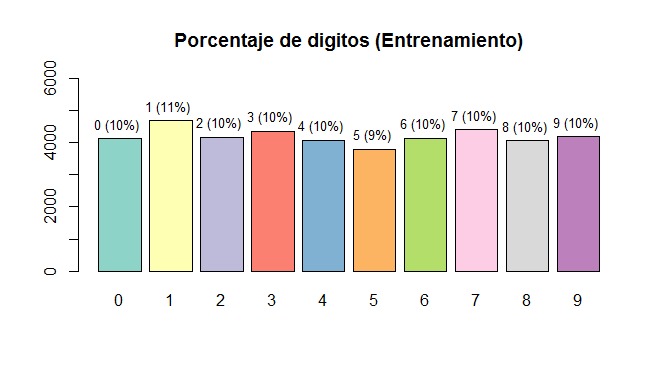
Las máquinas de soporte vectorial utilizan unas funciones matemáticas denominadas kernel, estas son usadas para convertir un problema que no tiene una solución lineal en un sistema de dos dimensiones a uno lineal en una dimensión más alta generando un hiperplano optimo que separa las clases, es óptimo en el sentido que aboga por minimizar el error de clasificación, esto teniendo en cuenta que para un mismo problema puede haber (y seguramente habrá) más de un hiperplano que me permita clasificar más observaciones.



**Figura 4.** Ilustración SVM

**Aplicación de la SVM al problema:** Se utilizó la librería *e1071* del software estadístico R para el entrenamiento de la máquina de soporte vectorial, se particiono el conjunto de datos en validación y entrenamiento, y se prepararon los datos para el análisis, se eliminaron algunas columnas, las cuales solamente contenían cero, y se escalaron los datos, esto con el fin de encontrar un mejor ajuste.

Primero, un pequeño análisis exploratorio de los datos:

 **Figura 5.** Frecuencia de los dígitos en el conjunto de entrenamiento.

De la figura 5 ose observa que se mantiene muy similar la frecuencia de todos los dígitos dentro del conjunto de datos que usaremos para el entrenamiento; esto es importante para reducir el sesgo.

Para el ajuste del modelo usamos la función SVM de la librería *e1071,* además se usó un 10% de los datos de entrenamiento y un kernel lineal, lo anterior fue decidido ya que inicialmente se usó una función para estimar el kernel y el parámetro C (el cual es una constante que representa el costo por restricciones) y dado el gran conjunto de datos no se llegó a una convergencia (en un tiempo optimo), la matriz de confusión obtenida es la siguiente:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **0** | 401 | 0 | 8 | 5 | 0 | 9 | 5 | 5 | 4 | 2 |
| **1** | 0 | 454 | 3 | 3 | 1 | 15 | 1 | 4 | 8 | 1 |
| **2** | 1 | 3 | 348 | 9 | 4 | 3 | 20 | 11 | 14 | 4 |
| **3** | 1 | 1 | 10 | 361 | 0 | 32 | 1 | 4 | 13 | 7 |
| **4** | 0 | 1 | 8 | 1 | 368 | 2 | 1 | 5 | 2 | 29 |
| **5** | 2 | 2 | 3 | 26 | 2 | 380 | 10 | 1 | 6 | 3 |
| **6** | 2 | 0 | 12 | 2 | 3 | 8 | 367 | 0 | 8 | 0 |
| **7** | 1 | 0 | 2 | 7 | 2 | 1 | 0 | 380 | 3 | 21 |
| **8** | 1 | 3 | 17 | 14 | 6 | 18 | 5 | 2 | 339 | 9 |
| **9** | 0 | 0 | 3 | 3 | 19 | 8 | 0 | 24 | 6 | 339 |
|  |  |  |  |  |  |  |  |  |  |  |

**Tabla 5.** Matriz de confusión modelo SVM

Un resumen del modelo es:

|  |  |
| --- | --- |
| **Parámetros:** |  |
| Ajuste | 0.87 |
| Tipo de SVM | C-Clasificación |
| Kernel | Lineal |
| C | 10 |
| gamma | 0.003968 |

**Tabla 6.** Resumen modelo SVM

De la tabla 5 se puede observar que el modelo hizo un gran trabajo tan solo utilizando el 10% de las observaciones dando un ajuste del 87%.

**Redes Neuronales**

Una red neuronal es un modelo de procesamiento de información usadas generalmente para el procesamiento de imágenes, reconocimiento de caracteres, predicciones de valores de acciones en la bolsa, préstamos bancarios, entre otros.  
 La red neuronal es un conjunto de unidades de entrada y salida de información que están interconectadas de forma que cada conexión tiene un peso asociado, estos pesos se ajustan (es decir, la red aprende) para predecir correctamente la clasificación de las entradas dadas.

Para realizar el modelo se empleó la librería *Keras* de R  
Los resultados se presentan a continuación mediante la matriz de confusión.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **0** | 5898 | 0 | 4 | 2 | 1 | 3 | 6 | 1 | 13 | 4 |
| **1** | 1 | 6710 | 10 | 1 | 6 | 1 | 0 | 5 | 18 | 2 |
| **2** | 7 | 11 | 5917 | 34 | 3 | 5 | 1 | 11 | 11 | 1 |
| **3** | 0 | 2 | 5 | 6041 | 0 | 11 | 0 | 3 | 11 | 12 |
| **4** | 0 | 4 | 4 | 0 | 5809 | 2 | 4 | 4 | 2 | 17 |
| **5** | 4 | 0 | 1 | 28 | 0 | 5375 | 6 | 1 | 22 | 26 |
| **6** | 5 | 2 | 6 | 0 | 4 | 14 | 5899 | 0 | 18 | 1 |
| **7** | 2 | 10 | 10 | 9 | 3 | 1 | 0 | 6229 | 4 | 22 |
| **8** | 3 | 2 | 0 | 10 | 1 | 2 | 2 | 2 | 5740 | 6 |
| **9** | 3 | 1 | 1 | 6 | 15 | 7 | 0 | 9 | 12 | 5858 |

**Figura 6.** Matriz de confusión.

Porcentaje de clasificación correcta:

Con una probabilidad del 99.13% una observación está clasificada correctamente.

Porcentaje de mala clasificación:

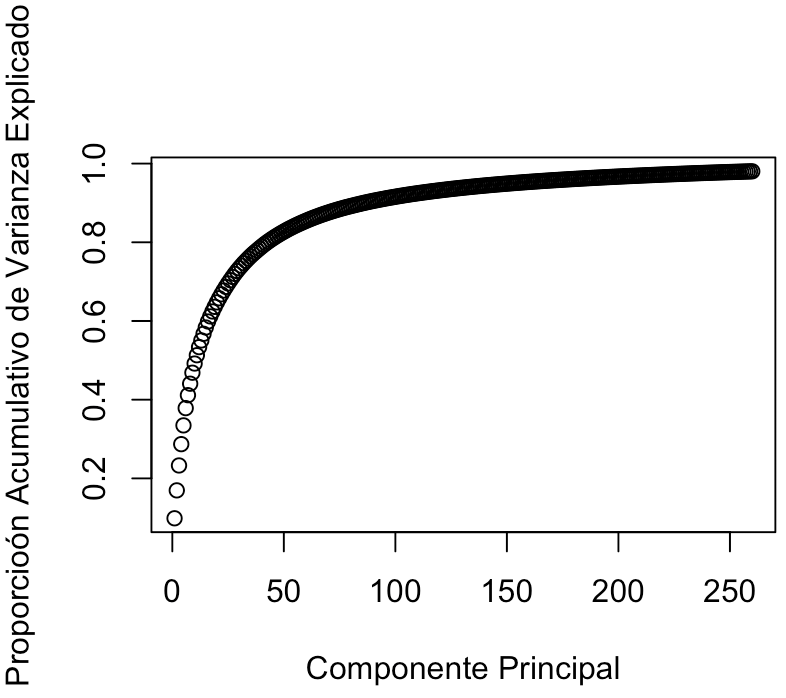
Hay una probabilidad del 0.87% de que una observación este mal clasificada.

Por tanto, se concluye que el modelo clasifica muy bien, puesto que la tasa de clasificación correcta es bastante alta.

**Arbol de Clasificación**

Aprendizaje basado en árboles de decisión utiliza un árbol de decisión como un modelo predictivo que mapea observaciones sobre un artículo a conclusiones sobre el valor objetivo del artículo. Los modelos de árbol, donde la variable de destino puede tomar un conjunto finito de valores se denominan árboles de clasificación. En estas estructuras de árbol, las hojas representan etiquetas de clase y las ramas representan las conjunciones de características que conducen a esas etiquetas de clase.

Usamos análisis de componentes principales (ACP) que es una técnica utilizada para describir un conjunto de datos en términos de nuevas variables ("componentes") no correlacionadas para construir el modelo predictivo. Luego, usamos la librería Rpart para ajustar el modelo con las nuevos componentes.



**Figura7**. Progreso de Varianza Explicado

Los resultados del modelo son los siguientes.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **0** | 211 | 0 | 29 | 68 | 0 | 30 | 14 | 5 | 25 | 0 |
| **1** | 0 | 392 | 16 | 13 | 0 | 28 | 7 | 0 | 2 | 1 |
| **2** | 4 | 6 | 286 | 30 | 1 |  | 16 | 7 | 45 | 7 |
| **3** | 1 | 2 | 9 | 325 | 2 | 33 | 7 | 3 | 46 | 2 |
| **4** | 0 | 4 | 13 | 7 | 221 | 4 | 9 | 7 | 7 | 113 |
| **5** | 34 | 1 | 31 | 83 | 15 | 107 | 14 | 15 | 39 | 8 |
| **6** | 1 | 2 | 46 | 24 | 0 | 15 | 300 | 7 | 1 | 23 |
| **7** | 4 | 18 | 7 | 6 | 8 | 1 | 2 | 306 | 28 | 29 |
| **8** | 0 | 2 | 19 | 40 | 3 | 40 | 4 | 12 | 240 | 15 |
| **9** | 0 | 9 | 6 | 10 | 57 | 9 | 6 | 53 | 16 | 226 |

**Figura8**. Progreso de Varianza Explicado

Porcentaje de clasificación correcta:

Con una probabilidad del 65.35% una observación está clasificada correctamente.

Porcentaje de mala clasificación:

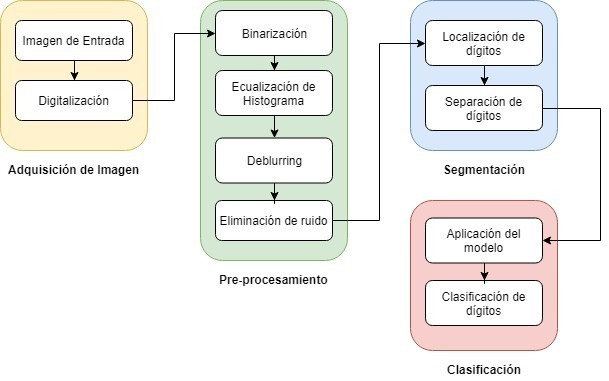
Hay una probabilidad del 34.65% de que una observación este mal clasificada

Según estos resultados podemos concluir que la prediccion del modelo no es muy buena ya que un 65.35% de certeza de clasificación no es muy convincente.

**Esquema de componentes**

Teniendo en cuenta los cinco métodos de clasificación mencionados anteriormente, el modelo que clasifica mejor es el modelo que utiliza redes neuronales, ya que supera ampliamente a los demás modelos en cuanto a clasificación correcta de observaciones.

Para este modelo hemos diseñado un esquema de componentes que permita explotarlo al máximo, el cual es el siguiente:



**Figura9**. Esquema de componentes

La arquitectura diseñada está compuesta básicamente de 4 componentes principales: Adquisición, preprocesamiento, segmentación, y clasificación, como se muestra en la Figura 9.

El primer componente es el de adquisición de la imagen. La imagen se puede obtener desde un escáner o cualquier otro origen electrónico. Esta será almacenada en cualquier formato de imagen, y esta imagen podrá estar a color, en escala de grises, entre otros.

El componente de preprocesamiento se encarga, entre otras funciones, de binarizar la imagen, es decir que sus posibles valores sean verdadero y falso que corresponden a los colores blanco y negro respectivamente. Este también se encarga de detectar y eliminar el ruido en la imagen, el cual puede ser causado por la calidad del papel, el escáner, etc.

El siguiente componente es el de segmentación, el cual toma la imagen ya procesada por el anterior componente, libre de ruido y binarizada. Este componente se encarga de localizar y separar los dígitos para su posterior clasificación. La salida de este componente es una lista de dígitos.

El último componente es el componente de clasificación, el cual su entrada es la lista de dígitos obtenidas por el anterior componente, a cada dígito de la lista se aplica el modelo que obtuvimos y se obtiene la predicción correspondiente.

De esta manera, podemos pasar una imagen como la entrada del sistema, como un código postal o una carta, y al pasar por todos los componentes, podremos obtener el contenido de esta (Los dígitos presentes en la imagen).

**Referencias:**

Wikipedia contributors. ( Febrero 22 del 2019). MNIST database. Recuperado de <https://en.wikipedia.org/wiki/MNIST_database>

Hastie, T. Tibshirani, R. Friedman, J.*The Elements of Statistical Learning.* Springer.

Allaire, I Joseph J., Chollet, Francois., (2018), Deep learning with R,

Marjani, A. (Octubre 9 del 2017). Digit Recognition with KNN, Neural Network, and SVM. Rpubs. Recuperado de https://api.rpubs.com/dardodel/digit\_recognition\_KNN\_NN\_SVM

A. Navlani.(2019, January 18) Neural Networks Models in R. Retrieved from: https://www.datacamp.com/community/tutoriales/neural-networks-models-r

Neha (2016, Diciembre 24) Digit Recognition. Recuperado de: https://rpubs.com/neha5/242005

Wikipedia contributors. ( Marzo 8 del 2019). MNIST database. Recuperado de https://es.wikipedia.org/wiki/Análisis\_de\_componentes\_principales