

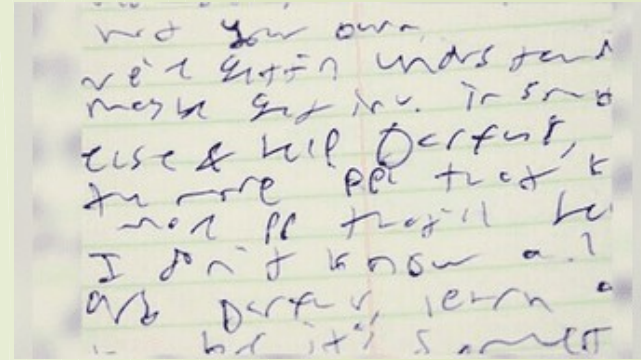


Proyecto Alfabeto

Por:

- Emilio Aced Fuentes
- Roberto Alcover Couso
- Arturo Blázquez Pérez
- Nicolás Trejo Moya

Problema

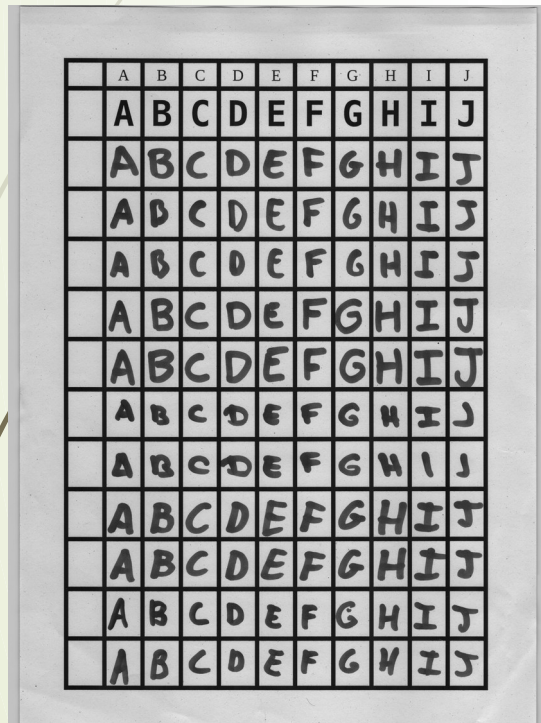


- ▢ Clasificación de letras:
 - ▢ ¿Es una I o una J?
- ▢ Se nos da una base de datos de letras manuscritas
- ▢ ¿Qué modelos usar?
- ▢ ¿Cómo entrenarlos?
- ▢ ¿A que damos prioridad al acierto o al tiempo?

Datos

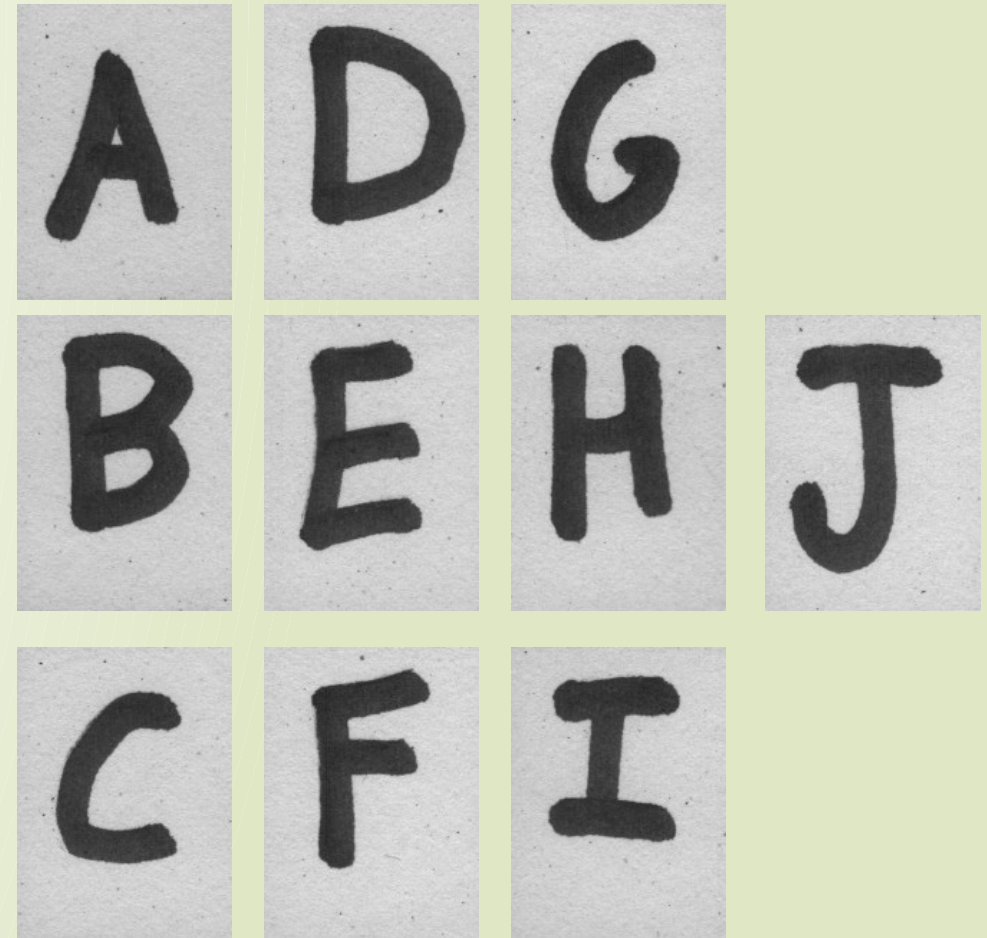
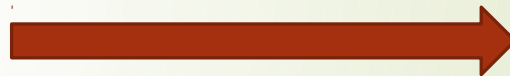
Obtención:

Datos en crudo



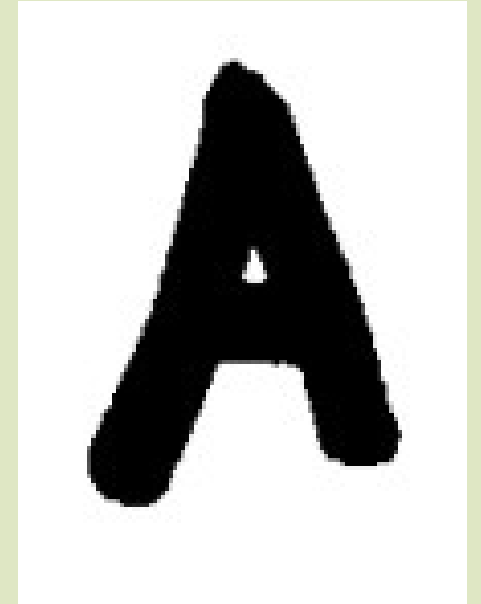
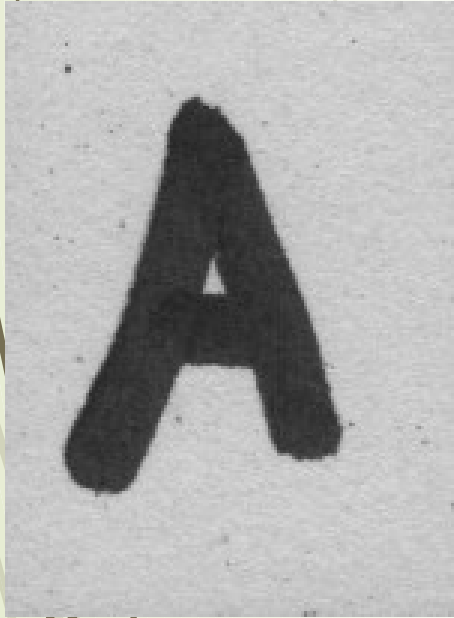
	A	B	C	D	E	F	G	H	I	J
	A	B	C	D	E	F	G	H	I	J
	A	B	C	D	E	F	G	H	I	J
	A	B	C	D	E	F	G	H	I	J
	A	B	C	D	E	F	G	H	I	J
	A	B	C	D	E	F	G	H	I	J
	A	B	C	D	E	F	G	H	I	J
	A	B	C	D	E	F	G	H	I	J
	A	B	C	D	E	F	G	H	I	J
	A	B	C	D	E	F	G	H	I	J
	A	B	C	D	E	F	G	H	I	J

Aplicación del .ipnb



Preprocesamiento

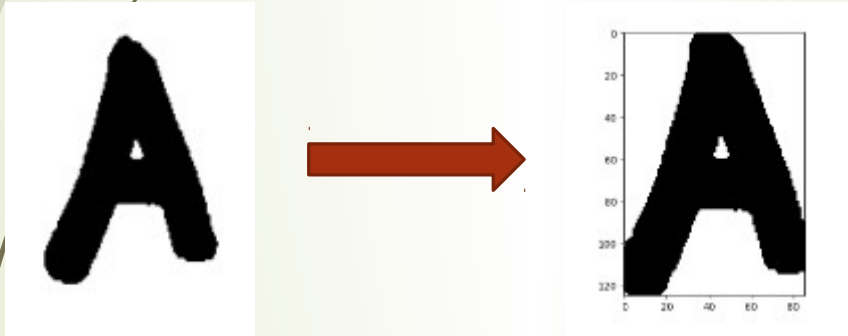
Imagen tras aplicar aplicar
umbralización de la imagen mediante
otsu y un filtro de mediana de tamaño 3



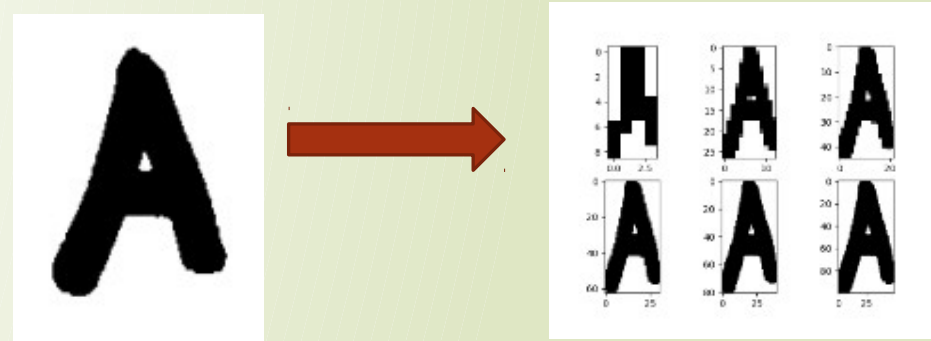
Atributos

- En un principio tenemos imágenes de tamaño 150x206.
- Necesidad de reducir el espacio de atributos.
- ¿Qué hacemos?

Reducimos el espacio de atributos recortando las zonas blancas dado que no aportan información.



También reducimos el número de atributos interpolando la imagen.



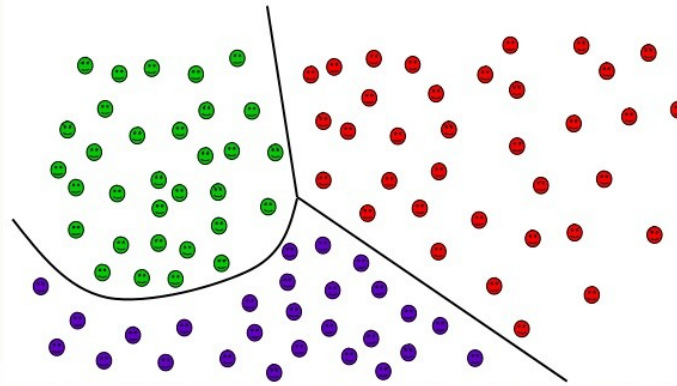


Entrenamiento

- ▮ Para el entrenamiento usaremos el 50% de los datos
 - ▮ RandomForest
 - ▮ Crea los árboles de decisión que decidirán por voto a que clase pertenece un ejemplo en la clasificación.
 - ▮ SCV
 - ▮ Es un clasificador lineal, por ello calcula el vector de pesos de cada clase.
 - ▮ KNN
 - ▮ Simplemente almacena los datos para después calcular las distancias a un ejemplo.

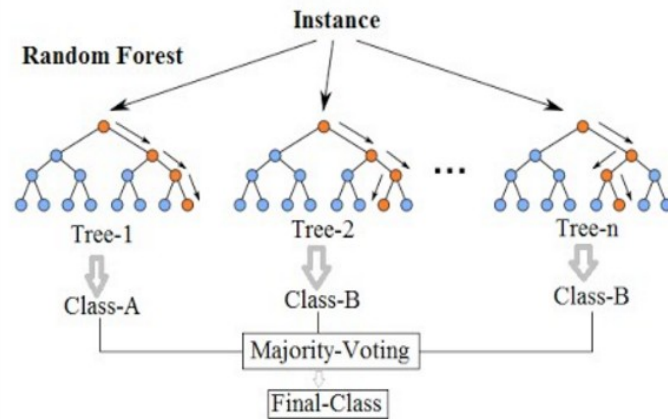
Modelos

Multi-class classification

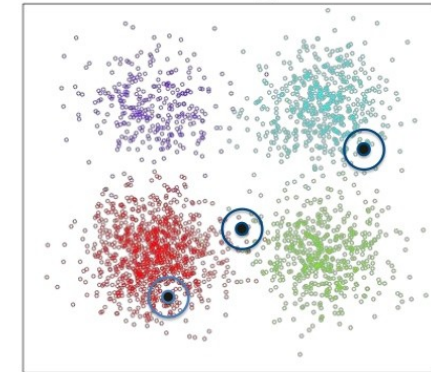


35

Random Forest Simplified



K-Nearest Neighbor Example



Random Forest

Scikit-learn:

Parámetros:

- Número de árboles: 500
- Profundidad: 10
- Tasa de acierto: 92.57%

Atributos:

- Interpolados
- Eliminación de columnas y filas poco significativas

Ejemplos de clasificación:



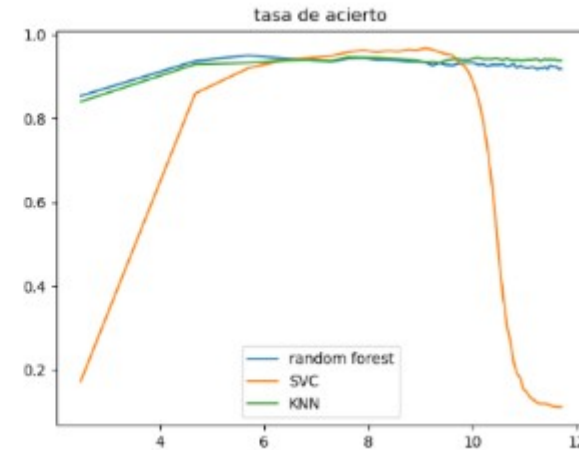
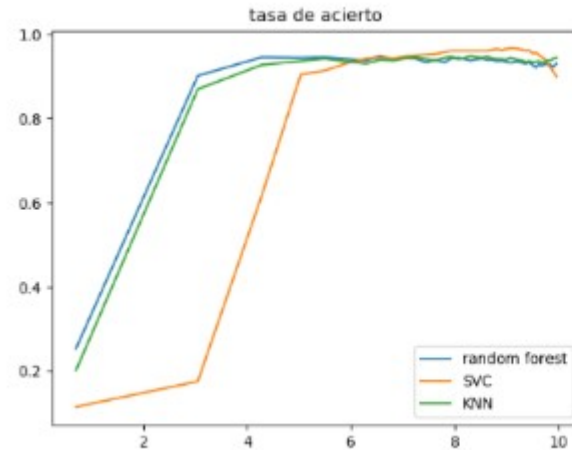
Matriz de confusión:

Clase	Precisión	Sensibilidad
A	0.93	0.99
B	0.86	0.75
C	0.97	0.99
D	0.91	0.92
E	0.95	0.92
F	0.88	0.98
G	0.96	0.97
H	1	0.93
I	0.88	0.87
J	0.90	0.93

$$M = \begin{bmatrix} 69 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 48 & 0 & 5 & 2 & 3 & 0 & 0 & 4 & 0 \\ 0 & 0 & 73 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 59 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 54 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 61 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 66 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 & 0 & 67 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 58 & 6 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 56 \end{bmatrix}$$

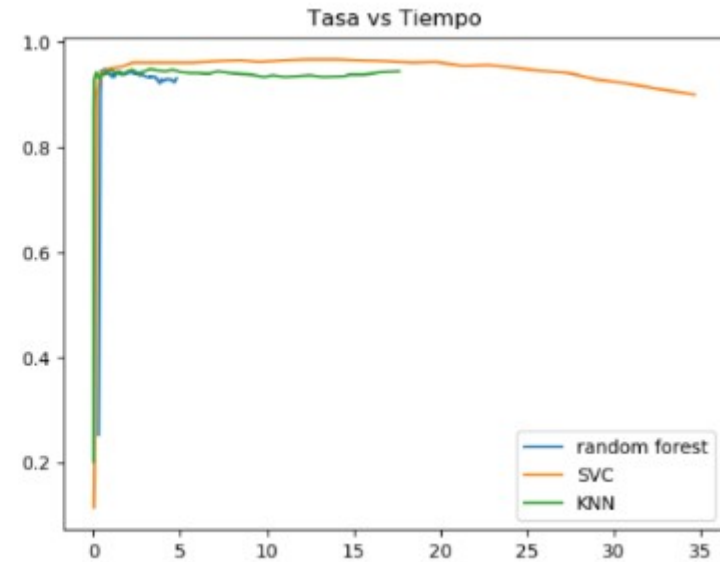
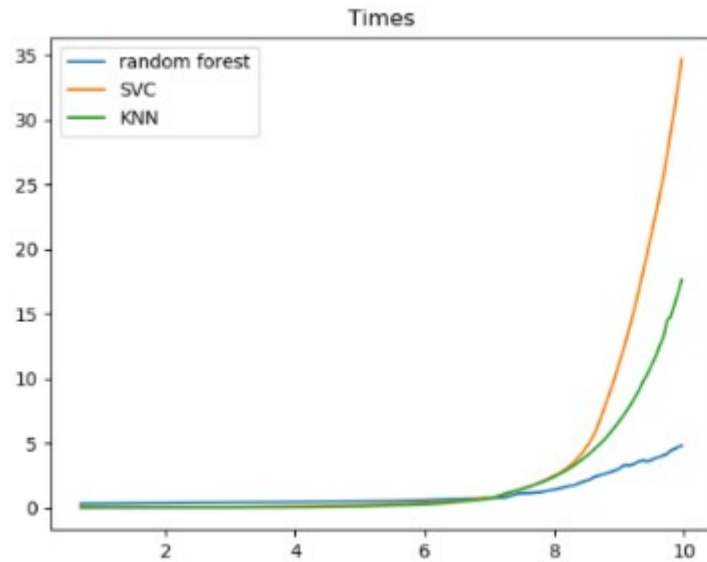
Número de atributos vs Tasa de

a



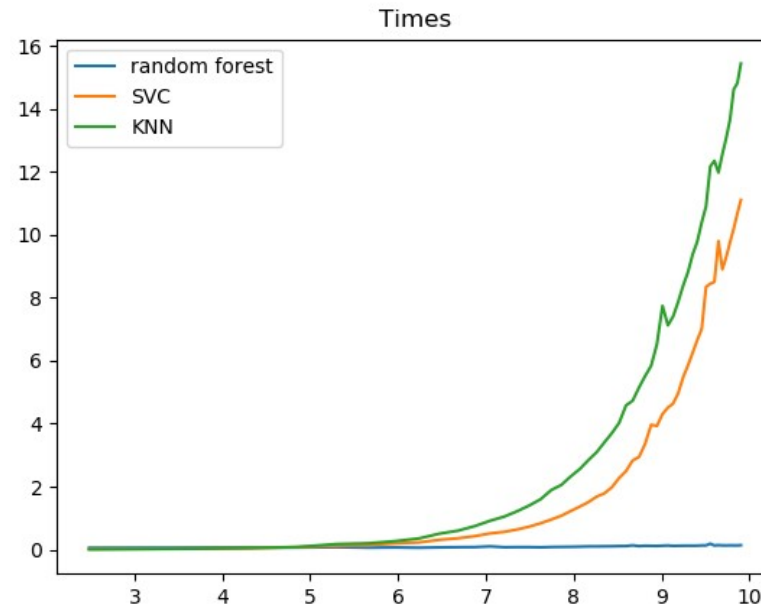
- ❑ SVC alcanza su máximo con imágenes de tamaño 100x50 falla al aumentar el tamaño de las imágenes.
- ❑ Random Forest y KNN trabajan bien con tamaños de imágenes menores.

Tiempo vs Tasa de acierto



- ❑ SVC es el más lento
- ❑ KNN tiene un rendimiento medio
- ❑ Random Forest es el más eficiente

Atributos vs Tiempo de clasificación



- ❑ En caso de tener tiempo ilimitado siempre escogeremos a SVC ante KNN dado que SVC tiene menor tiempo de clasificación y mayor tasa de acierto
- ❑ En limitaciones de tiempo siempre escogeremos RandomForest



Conclusiones

- ❑ SVC clasifica muy bien pero no es nada robusto respecto al número de atributos
- ❑ Random Forest es muy rápido pero tiene menor tasa de acierto que SVC y KNN
- ❑ KNN es el segundo clasificador en tasa de aciertos y tiempo