Resultados del Análisis de Modelos de Machine Learning para Reconocimiento de Dígitos

Introducción:

Componentes del grupo de trabajo:

- Isidro Javier García Fernández
- Álvaro Sánchez Hernández
- José Antonio Luque Salguero
- Jesús Escudero Moreno

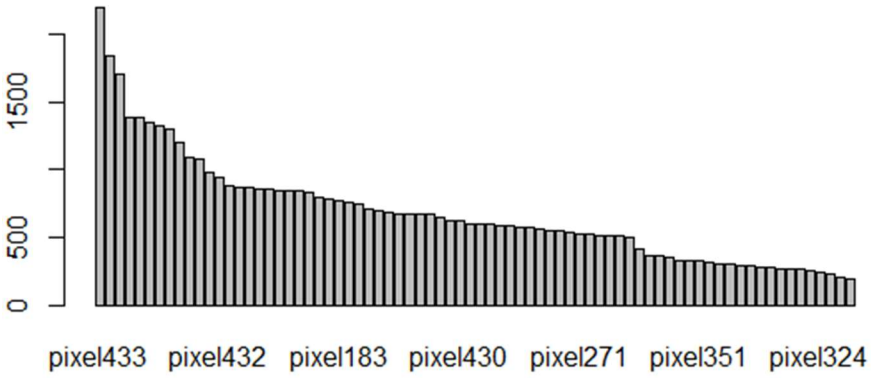
En esta práctica, se ha abordado el problema de reconocimiento de dígitos mediante la aplicación de varios modelos de machine learning. Se han utilizado enfoques como árboles de decisión, bosques aleatorios, SVM, bagging y boosting. Se evalúa el desempeño de cada modelo utilizando matrices de confusión y se calcula la precisión.

Datos:

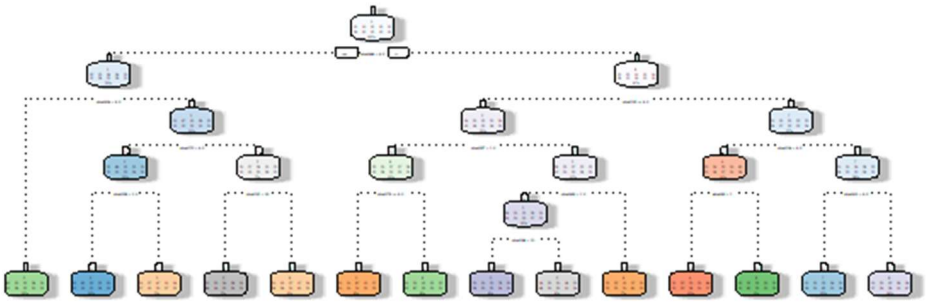
Se cargaron los datos desde archivos CSV procedentes de https://www.kaggle.com/competitions/digit-recognizer/data , y se realizaron ajustes necesarios, como la conversión de etiquetas a factores y la validación cruzada para dividir el conjunto de entrenamiento y prueba.

Árbol de Decisión (rpart):

Se construyó un árbol de decisión utilizando el método rpart. Se muestra una matriz de confusión y se calcula la precisión del modelo. Podemos observar la importancia que da este modelo de predicción a cada variable, siendo la más predominante el Pixel 433.



Además, se presenta un gráfico del árbol. Consigue una tasa de acierto de 64,16 %



Random Forest:

Se implementaron varios modelos Random Forest con diferentes números de árboles. Se muestran las matrices de confusión y precisiones para 10, 5 y 2 árboles respectivamente.

- Con 10 árboles consigue una tasa de acierto de 91,65 %
- Con 5 árboles consigue una tasa de acierto de 87,82 %
- Con 2 árboles consigue una tasa de acierto de 77,97 %

SVM:

Se aplicaró un modelo SVM con kernel polinómico. Se detallan las matrices de confusión y las precisiones para ambos kernels, así como para un modelo SVM básico.

KSVM con kernel polinómico de grado 3 con 3 pliegues para validación cruzada consigue un accuracy de 97,33 %

Bagging:

Se utilizó el método de bagging para construir un modelo. Bagging construye varios modelos base (en este caso árboles de decisión) entrenados en subconjuntos aleatorios de datos y luego combina sus predicciones.

Se presenta la matriz de confusión y la precisión del modelo AdaBag. Se especifica un total de 9 modelos base y se configuran parámetros adicionales para el árbol de decisión. Por ejemplo se establece un $cp = 0.001$ que va relacionado con la complejidad del árbol, pues un valor elevado para este atributo provocaría una poda más agresiva para los árboles generados. En este caso se permite un mayor desarrollo de los mismos. También se establece un $minsplit = 7$, que establece el número mínimo de observaciones que deben existir en un nodo antes de que se considere la división. Un valor elevado para este atributo provocaría divisiones en nodos con muy pocas observaciones.

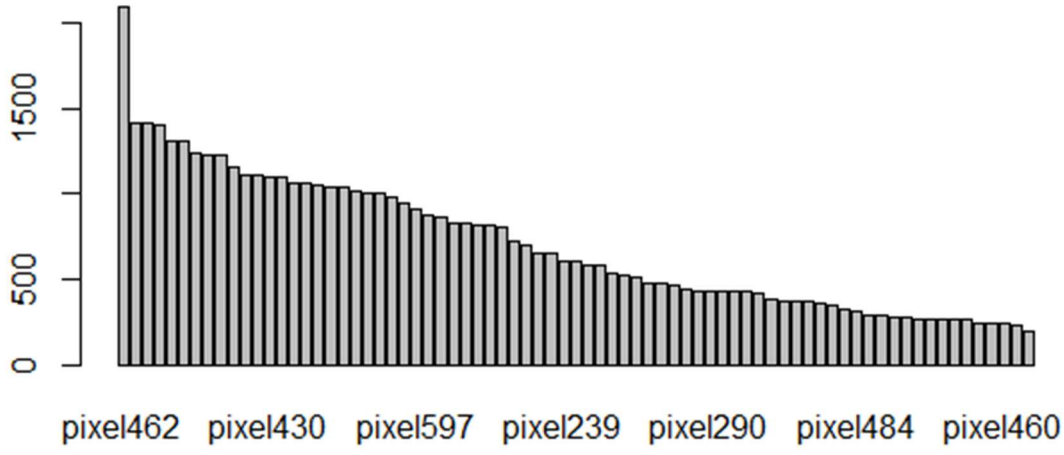
Se consigue una tasa de acierto de 85,71 %

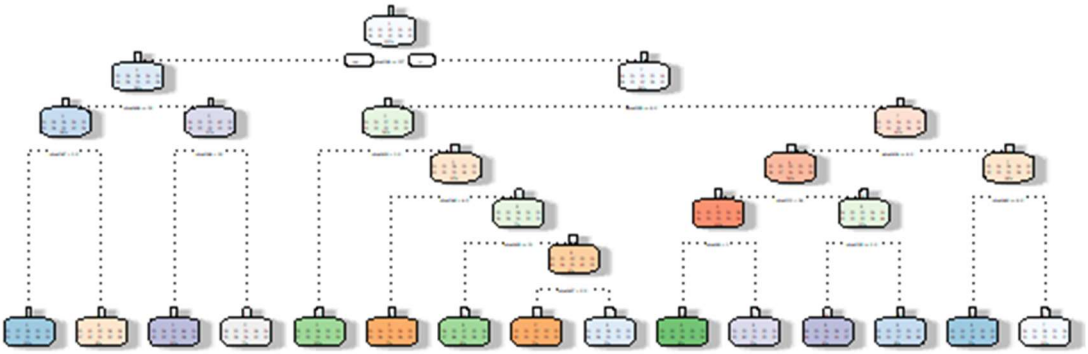
Boosting:

Se implementó el algoritmo de boosting (AdaBoost), que asigna pesos a las instancias de entrenamiento, dándole más énfasis a las instancias mal clasificadas en iteraciones anteriores. Se proporciona un resumen del modelo, así como la matriz de confusión y precisión resultante. Se realizan 10 iteraciones y se utiliza un coeficiente de aprendizaje según el método de Breiman (Random Forest).

Consigue una tasa de acierto de 77,89 %

Se presenta un gráfico del primer árbol que devuelve el modelo, junto con una gráfica con la importancia de los atributos para este modelo. En esta ocasión es más relevante el pixel 462.





El árbol escogido consigue una tasa de acierto de 62,80 %

Conclusión:

Se ha explorado y evaluado el rendimiento de diversos modelos de machine learning en el reconocimiento de dígitos. Los resultados proporcionan información valiosa sobre la eficacia de cada enfoque y sirven como base para la selección del modelo más adecuado para este problema específico. Ha quedado demostrado en este pequeño estudio que el modelo que mejor ha clasificado ha sido el de KSVM con Kernel polinomial de grado 3, y el que peor ha clasificado ha sido el primer árbol escogido del modelo de Boosting, aunque hay que tener en cuenta la complejidad temporal y espacial de cada uno de los modelos para poder decidir qué nos puede interesar.