1 Random forest(RF)

Se usa esta técnica puesto que es un muy buen clasificador. En este caso al meter información extra a el dataset tendremos una cantidad grande de variables y esta técnica trabaja muy bien cuando se tienen gran numero de variables. En este caso se ha añadido información de grado 2.

1.1 Especificación de parámetros RF

En este caso nos hemos centrado en la elección de una buena cantidad de arboles y la cantidad de variables que utilizara cada árbol para predecir. Para seleccionar estos parámetros se han realizado varias pruebas con la partición de validación. Se ha medido el error con la partición de validación con tres cantidades distintas de variables. El numero de variables elegidas para cada una son las siguientes:

1. Raíz: En este caso se tendrá una cantidad de características para cada árbol igual a la raíz cuadrada del numero de características.
2. Mitad: La cantidad de variables sera en esta opción la mitad de las características.
3. Cantidad de características: Para esta opción cada árbol utilizara todas las características.

El numero de arboles lo iremos aumentando de 10 en 10 hasta llegar a 250.
Pasamos a mostrar el grafico para analizar el mejor numero de características para utilizar.

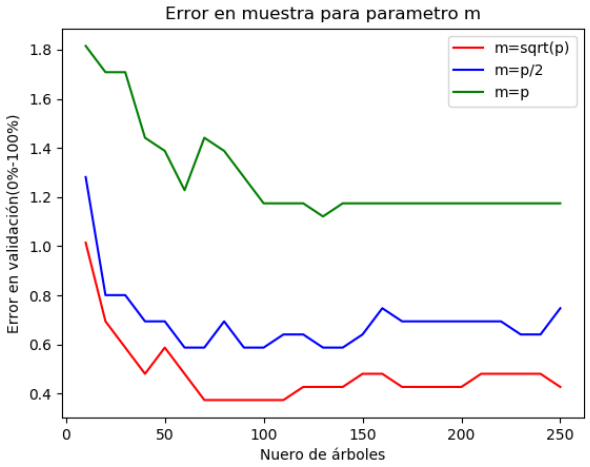

Figure 1

En el grafico se ve claramente que la mejor opción es la raíz. También podemos sacar de este grafico cual es la mejor opción para el numero de arboles necesarios. En este caso a partir de 70 se tiende a igualar y no mejora. Para decidirnos por el numero de arboles tendremos en cuenta también el siguiente grafico, donde se muestra el tiempo de ejecución para cada cantidad de arboles.

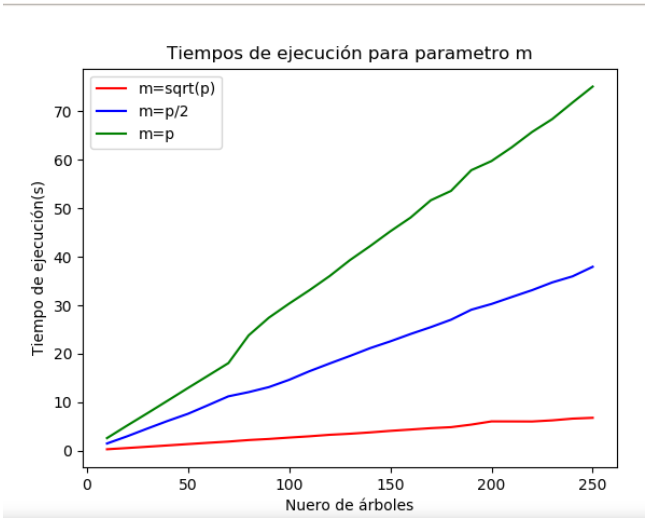

Figure 2

En este caso se ve que con mas cantidad de arboles tendremos una mayor tiempo de ejecución. Por tanto he decido definir la cantidad de arboles a 100 y quedarme con la opción de la raíz para el numero de características por árbol.

1.2 Valoracion de resultados.

La métrica utilizada en este caso es mean accuracy y el error obtenido en la partición de test es el 3.5162 % en un tiempo de ejecucion de 2.5 segundos. En este caso no es un mal porcentaje de error pero vamos a analizar la matriz de confusión para tener una mejor idea.

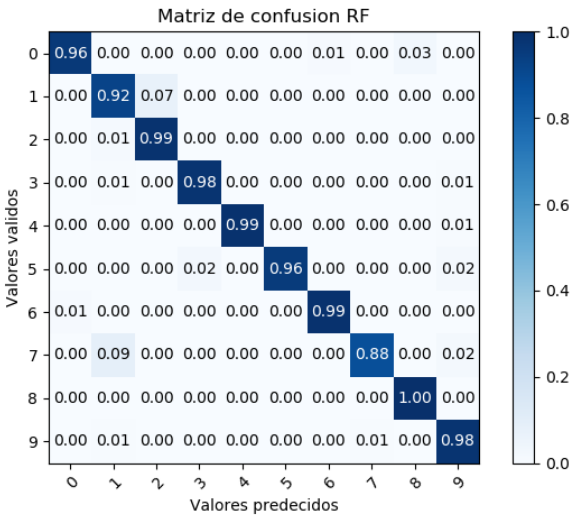

Figure 3

En este caso se ve que tenemos falla mucho al prediciendo 7 como 1 y también confunde los 1 con 2. Esto nos podría llegar a fallar mas de lo común en el reconocimiento de estos dígitos.

En este caso he querido utilizar una partición de validación en vez de utilizar el método de oob para la validación, por realizarlo de la misma manera que los modelos anteriores.