El machine Learning resulta complejo de explicar más que nada para aquellas personas o clientes que no tienen conocimiento a cerca de el, y si ya es difícil por si solo, ahora con todas las configuraciones de hiper parámetros que vimos más el cómo afecta a cada algoritmo lo vuelve más complicado aun. Al estar desarrollando los algoritmos vimos que hay aquellos que se pueden ejecutar de forma fácil como lo es SVM gracias al grid\_search.best\_estimator\_ que nos ahorra mucho trabajo, sin contar que es de los pocos que no requieren tanto tratamiento con los datos, sin embargo, softmax se puede ejecutar con solo 4 renglones, pero como mencione con anterioridad, este depende de que el data set este limpio, ya sea transformar las variables categóricas a numérica so imputarlas.  
  
Ya cuando hablamos de árboles de decisiones, estos son más fáciles de explicar, para gente con conocimiento en informática se podría decir que funciona como un if/else por cada rama hay otra y para los clientes, se podría tomar la idea de escoger caminos cada que nos encontremos en un punto y no sepamos a donde ir, por así decirlo.  
  
Con respecto a BaggingClassifier  y ada\_reg.predict se me hicieron de los algoritmos más fáciles de programar e interpretar. No son complejos, aunque requieren de un pre procesamiento de datos o bien dejar solo la variable categórica a predecir en caso de existir más .   
  
Todos los algoritmos estuvieron en el 30 de precisión, sin embargo, los random forest alcanzaron 1 en el train y softmax se quedó a nada de llegar a 1.  
  
Cual algoritmo escoger dependerá de nuestro objetivo, numero de variables y el tipo de estas, pues habrá mucho que hacer con unos, pero darán mejor resultado y otros requerirán del ajuste de hiper parámetros