**Diferencia entre Algoritmos**

1. **SVM.SVC con 43410 registros y kernel='linear' (3,3 hrs 36% accuracy)**

**65% de comentarios etiquetados.**

1. **SGDClassifier con 43410 registros y loss='log' (8,6 min 30% accuracy)**

**48,4% de comentarios etiquetados.**

1. **RandomForestRegressor con 43410 registros y n\_estimators=50 (1,5 hrs 32,3% accuracy) 76,5% de comentarios etiquetados.**
2. ¿De qué manera se puede complementar la solución? Pensar en propuestas para el cliente.

**Una forma de complementar la solución, sería aumentando el valor del modelo de clasificación de texto. Esto quiere decir, alojar los modelos de machine learning en un sistema web en la nube, que, a pesar de requerir un esfuerzo relativo mayor, entregaría muchas más funcionalidades. Por ejemplo, la posibilidad de permitir al usuario elegir el modelo a utilizar, conociendo a priori el tiempo de ejecución, certeza y el porcentaje de etiquetado que entrega cada modelo, la posibilidad de realizar una cascada de los datos a través de los modelos, comparar entre los resultados de los modelos o acceder a funcionalidades de visualización de resultados.**

1. ¿Cómo se podría simplificar la tarea?

**El modelo podría simplificarse eludiendo el preprocesado de texto, etapa que requiere de mucho tiempo si el proceso se realiza con grandes cantidades de datos. Sin embargo, la cantidad de tiempo que demora el entrenamiento, sería mayor, y probablemente, los resultados serían menos exactos. Otra forma podría ser no utilizar multietiquetado, para no confundir al modelo (solo trabajar con clases), o bien, reduciendo la cantidad de etiquetas, lo que por un lado permitiría mejores niveles de asertividad, lo que lamentablemente haría perder el nivel de detalle de emociones que presenta el dataset.**

1. ¿Cuáles pueden ser las limitaciones, riesgos, sesgos de los modelos al implementar este tipo de soluciones?

**Las limitaciones de estos modelos pasan por ser soluciones generalmente costosas y que pueden llevar mucho tiempo dada la binarización de los datos tipo texto y el multietiquetado. En este caso en particular, además, el multietiquetado contenía 28 valores distintos, por lo que la cantidad de veces que el código es aplicado, aumenta. Por otra parte, estos modelos presentan dos situaciones que aumentan el riesgo de su uso; 1) Los modelos no clasifican a todos los comentarios, por ende, se estarían tomando decisiones sin la totalidad de los datos. 2) El sesgo que puede presentar este modelo, ya que pueden existir errores en los datos, cosa que se infiere en el enunciado del desafío: “La empresa ya ha clasificado manualmente alrededor de 48 comentarios, con 28 emociones diferentes”, sin embargo, un poco más arriba, la empresa evidencia que “Actualmente, los comentarios se clasifican manualmente, lo que es un proceso lento y propenso a errores”, lo que podría conllevar a que el resultado final, necesariamente presente sesgo (error en la captura de datos). Por otra parte, también puede ser posible que estos datos presenten sesgo algorítmico, sesgo que afecta principalmente a las menorías o a aquellos grupos de datos que no están bien representados. Los modelos pueden basarse en gran medida en etiquetas que están correlacionados con otras etiquetas y asignar un mal etiquetado a un comentario.**

1. ¿Qué otras cosas hay que considerar al momento de implementar un proyecto como este?

**Primordial en primer lugar es realizar una buena preparación de los datos para que no tengamos que descartar el modelo por malos resultados. Es importante eliminar ciertos caracteres y palabras clave (stopwords). Además, es importante utilizar diferentes funciones para algunos procesos, como por ejemplo CountVectorizer vs TfidfVectorizer, para elegir la que mejor se ajusta a los datos de entrenamiento (en este caso, TfidfVectorizer emplea ponderación inversa, es decir, el modelo otorga menor ponderación a las palabras con menor frecuencia, aprendiendo que estas tienen menor importancia). Por otra parte, también es necesario practicar con diferentes modelos de machine learning y jugar con sus parámetros, para ver qué algoritmo se ajusta mejor a los requerimientos del cliente (mayor velocidad, mayor precisión o mayor exhaustividad).**

1. **SVM.SVC con 3000 registros y kernel='linear' (59 seg 30% accuracy) 1895**
2. **SGDClassifier con 43410 registros (429 seg 28% accuracy)**

**RandomForestRegressor con 3000 registros y n\_estimators=10 (381 seg 32,8% accuracy)**

**RandomForestRegressor con 3000 registros y vv (761 seg 32,5% accuracy)**

**SGDClassifier con 3000 registros (5 seg 29% accuracy)**

**LogisticRegression con 3000 registros (12 seg 32,9% accuracy)**

0 0.71 0.46 0.56 826

1 0.75 0.70 0.72 441

2 0.55 0.02 0.03 388

3 0.77 0.19 0.30 158

4 0.57 0.31 0.40 52

5 0.92 0.12 0.21 189

6 0.52 0.30 0.38 108

7 0.93 0.86 0.90 525

8 0.00 0.00 0.00 19

9 0.56 0.46 0.50 284

10 0.72 0.65 0.68 381

11 0.00 0.00 0.00 35

12 0.56 0.22 0.31 313

13 0.70 0.39 0.50 328

14 0.32 0.32 0.32 19

15 0.55 0.10 0.17 214

16 0.00 0.00 0.00 30

17 0.53 0.60 0.56 99

18 0.79 0.25 0.38 273

19 0.55 0.36 0.44 225

20 0.59 0.60 0.60 2940

21 0.83 0.03 0.06 451

22 0.59 0.06 0.11 583

23 0.67 0.01 0.02 214

24 0.72 0.05 0.10 247

25 1.00 0.04 0.08 435

26 0.56 0.35 0.43 115

27 0.92 0.08 0.15 260