Rodrigo Muñoz Guerrero (A00572858)
Concentración IA Avanzada

Entrega portafolio de análisis

En el actual documento se hace el análisis y reporte sobre el desempeño del modelo elegido (Modelo con framework). A continuación se listaran las subcompetencias y links a la ubicación donde se evidencían. Así como debajo se encuentra el reporte.

SMA0104A

- Evalúa el modelo con un conjunto de prueba y un conjunto de validación
    - La implementación de la separación de dataset se encuentra en este código.
    - El análisis y explicación escrita de la separación se encuentra en esta parte del reporte.
- Detecta correctamente el grado de bias o sesgo: bajo medio alto
    - El análisis escrito del bias se encuentra en esta parte del reporte.
- Detecta correctamente el grado de varianza: bajo medio alto
    - El análisis escrito de la varianza se encuentra en esta parte del reporte.
- Explica el nivel de ajuste del modelo: underfit fitt overfit
    - El análisis escrito del ajuste del modelo se encuentra en esta parte del reporte.

Rodrigo Muñoz Guerrero (A00572858)
Concentración IA Avanzada

Reinheit Algorithmus

Modelo de regresión logística

Contexto

Este documento reporta el proceso de desarrollo y resultados del modelo 'reinheit algorithmus'; Algoritmo de regresión logística (o clasificación) que predice si una cerveza es de estilo 'IPA' o no, basándose en los atributos de la bebida *ABV* (Alcohol By Volume) y *IBU* (International Bitterness Unit).

El código de la implementación se puede consultar en este repositorio de Github.

Data

Los datos usados para entrenar y probar el modelo fueron obtenidos de Kaggle (https://www.kaggle.com/datasets/nickhould/craft-cans) y descargados en formato *csv*.

La estructura de la tabla de datos usada ('beers.csv') es la siguiente:

abv	(float)
ibu	(float)
id	(int)
name	(string)
style	(string)
brewery_id	(int)
ounces	(float)

Se decidió usar solamente las variables 'abv', 'ibu' y 'style' porque las demás variables no son representativas para el modelo ya que son nombre, identificadores numéricos o no influyen en las propiedades de la cerveza.

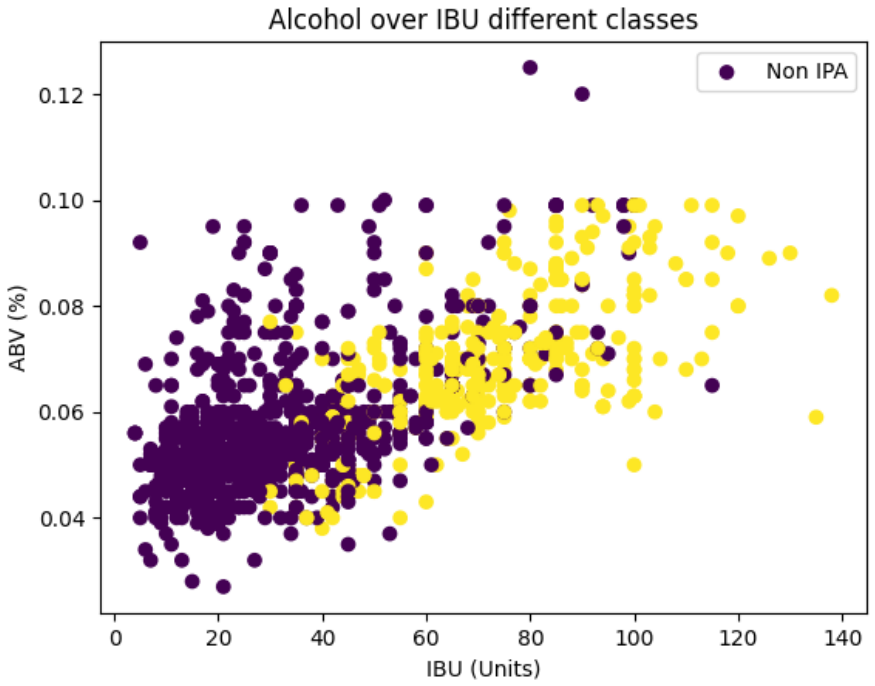
Los datos usados son entonces [ibu, abv], como las variables independientes (o las X) y [style], como la variable dependiente (o la y, que funge como las etiquetas de clase).

Proceso de transformación de la data

Como se mencionó anteriormente, se usaron los datos de *ibu* y *abv* y lo primero que se hizo, fue quitar los registros que dichas columnas contenían datos vacíos (NaN) porque al ser los únicos dos parámetros, es crucial tener la información completa.

Lo siguiente es transformar nuestras etiquetas y asignar el valor de "IPA" a toda etiqueta que contenga el mismo valor (Esto para homogeneizar las variaciones del estilo como Imperial IPA, Double IPA, Session IPA, etc.) y después, encodificar las etiquetas a números, siendo 0 "No IPA" y 1, "IPA".

Vemos la distribución de nuestros datos según su etiqueta en la siguiente gráfica.



(fig. 1, All data plotted by class)

Observamos que el dataset tiene mucho ruido al tener registros con parámetros similares a una IPA pero siendo de un estilo diferente, lo cuál podría dar una alta varianza al modelo.

Teniendo el dataset listo, dividimos el dataset en dos partes, una para entrenar el modelo, y otra para probar el modelo. La proporción de la división depende de qué tan grande sea el dataset y si se quiere tener una parte extra para validación y ajuste de hiperparámetros, en este caso se divide el dataset en dos partes, entrenamiento (train) y pruebas (test); Con una proporción de 80/20 respectivamente.

Es importante que al dataset se le altere el orden (shuffle) de los registros para que el modelo no aprenda un patrón o que no haya una buena distribución de las diferentes clases en el dataset y así, evitar un under u overfit. Además, se estandarizan los datos con un *Standard Scaler* de sklearn para que le sea más fácil al modelo ya que tenemos datos porcentuales entre 0 y 1 (ABV), así como datos entre 0 y 140 (IBU).

Modelo

El modelo es de tipo regresión logística o también conocido como de clasificación el cuál, como el nombre lo dice, es utilizado para predecir a qué “clase” o “etiqueta” pertenece la información sometida a predicción. Como se mencionó anteriormente, el modelo predecirá un resultado binario de la siguiente clasificación:

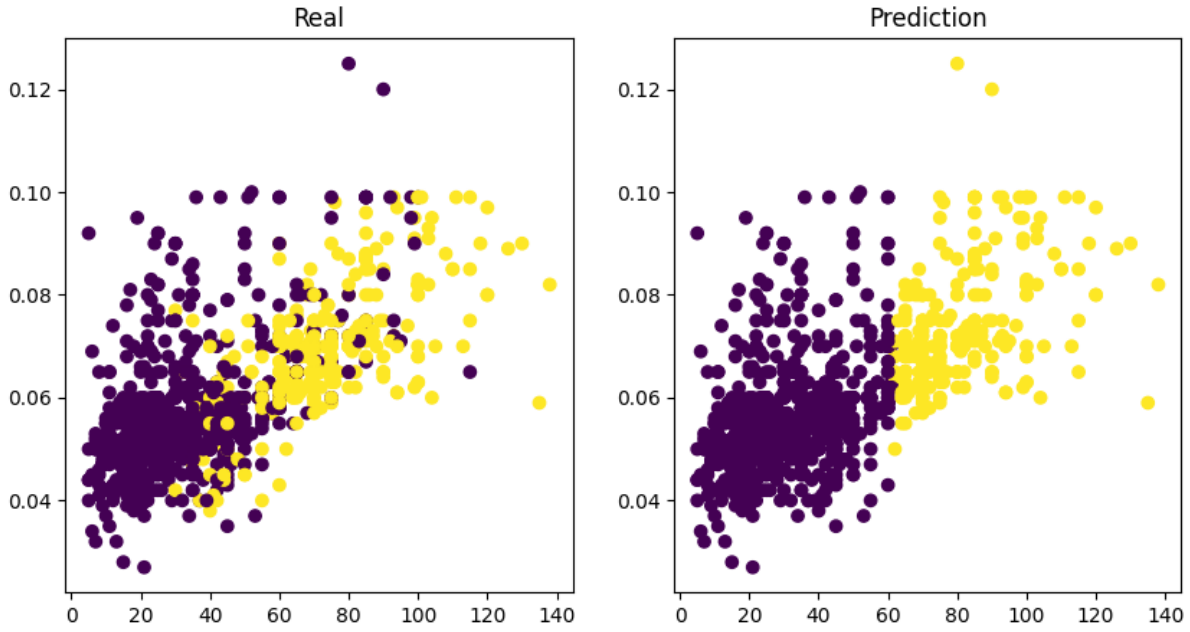
- 0 (cero) = Una cerveza de estilo diferente a IPA
- 1 (uno) = Una cerveza de estilo IPA (incluyendo variaciones)

Se utilizó la librería *sklearn* para el escalador, el modelo, la separación en sets de entrenamiento y prueba y para la matriz de confusión. No se definieron hiperparámetros iniciales para el modelo ni el entrenamiento.

Resultados y Evaluación

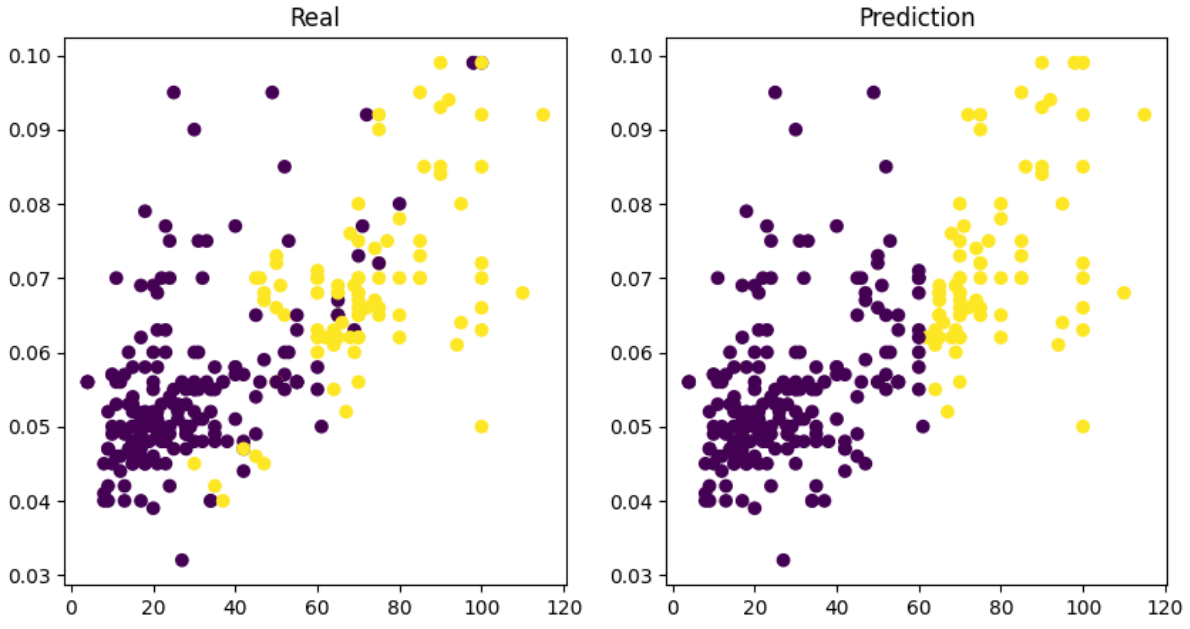
Utilizando el modelo anterior, se predijeron las clases o etiqueta de la data de prueba y concluyó en los siguientes resultados:

Entrenamiento



(fig. 2, Clasificación de datos reales y predecidos entrenamiento)

Pruebas



(fig. 3, Clasificación de datos reales y predecidos pruebas)

Como se puede observar, el modelo genera una clara *decision boundary line* en aproximadamente 62 IBUs, dejando fuera algunos registros que probablemente son de estilo "Session IPA", para el cuál los IBUs y el ABV disminuye significativamente del estilo base, entre otros, lo cuál genera un bias medio-alto.

Para poder concluir correctamente el desempeño del modelo, se calcula una "matriz de confusión" de la cuál podemos extraer información valiosa como la precisión y F1 que se explicará más adelante. Usamos la biblioteca *sklearn* para calcularla y obtenemos lo siguiente:

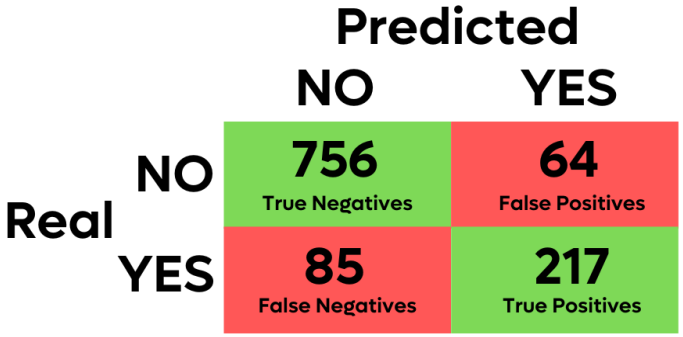
Entrenamiento = [[756, 64], [85, 217]]
Pruebas = [[178, 13], [23, 67]]

O mejor representado:

Entrenamiento

Precisión= 86.72%

F1 = 74.44%

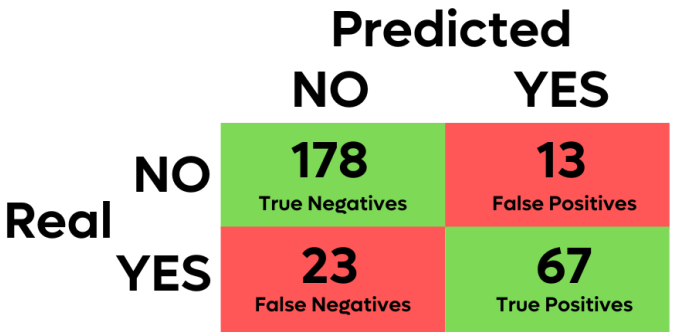


(fig. 4, Matriz de confusión de entrenamiento)

Pruebas

Precisión= 87.18%

F1 = 78.82%



(fig. 5, Matriz de confusión de pruebas)

Con la matriz de confusión anterior se calcula la precisión del modelo con la fórmula *(True Positives + True Negatives) / (Total).* Lo que le da al modelo una precisión de **87.2%**, dicho valor es igual a si se usa la librería *sklearn* con la función *score().* La precisión es una de las medidas más utilizadas para medir el desempeño de un modelo ya que el valor es porcentual y es más fácil de comprender sin el conocimiento de más información.

Otra medida de desempeño del modelo es el *F1 score*, el cual describe un porcentaje de la precisión y la exhaustividad (la cuál describe cuántas veces el modelo predijo correcta y positivamente del total de positivos reales existentes en el dataset). El *F1 score* se calcula con la fórmula *(2 * True Positives) / ((2 * True Positives) + False Positives + False Negatives* el cuál aplicado con la matriz de confusión, calculamos que el F1 score del modelo es **78.82%**. Se recuerda que tener un 100% de F1 score significa tener perfecta precisión y exhaustion.

Con los datos de precisión y F1 tanto de entrenamiento como de pruebas, se concluye que se tiene un buen puntaje en los dos y el modelo es viable para predecir estilos de cerveza IPA. Al tener una precisión similar en el entrenamiento y en pruebas, se descarta un overfit a los datos de entrenamiento sin embargo, visualmente se aprecia un underfit ya que la decision boundary line está muy marcada cuando en los datos reales, no lo está. En conclusión técnica, la data tiene demasiado ruido y se

necesitaría información adicional y un modelo más complejo para tener una clasificación de calidad para estilos de cerveza.

Como experto en estilos de cerveza, concluyo que el modelo es aceptable ya que existen otros estilos con propiedades parecidas a las de las IPA y para poder hacer una distinción entre ellos, se necesitan más datos como el *SRM* (Standard Reference Method) que describen el color de la cerveza; Por lo que éste modelo con éstos datos, es suficientemente bueno aunque podría hacerse un modelo mucho mejor teniendo datos adicionales.