FACULTAD DE CIENCIAS

Investigación Operativa

# ¿Qué vamos a comer hoy?

Modelos e Inteligencia Artificial


**Presentado por:**

Miguel Ballesteros Morilla
María Montero Díaz
Alejandro Morales Miranda
Sergio Torremocha Naranjo



UNIVERSIDAD DE MÁLAGA

Málaga, Octubre 2024

# Índice

# Introducción.

Para este proyecto, hemos enfocado nuestro problema en uno de los principales usos de la inteligencia artificial, la toma de decisiones.
Por tanto, hemos desarrollado un modelo de inteligencia artificial basado en la regresión logística multinomial con el objetivo de ofrecer recomendaciones personalizadas sobre opciones alimenticias.

Partiendo de dos variables cuantitativas, la edad y el presupuesto, el modelo sugiere entre cuatro tipos de cocina: italiana, española, asiática y fast food. La investigación se centra en analizar la relación entre estas variables y los gustos, buscando patrones que permitan generar sugerencias más precisas.

A través de datos reales y modificaciones en el modelo, con la intención de mejorar su funcionamiento, nuestro modelo busca mejorar la experiencia de los usuarios, garantizando que la personalización sea óptima y la propuesta culinaria del modelo se ajuste a las preferencias de estos.

# Capítulo 1

# Elección del problema a estudiar.

El problema que abordamos en este trabajo se enmarca dentro de una cuestión cotidiana que muchos se plantean en el momento de comer o cenar: "¿Qué o dónde comemos?". Sin embargo, el enfoque de nuestro problema no se centrará en proporcionar una respuesta exacta a esta pregunta, sino que, a partir de los datos obtenidos de una persona y utilizando información previamente recopilada de otros usuarios, generará una recomendación personalizada. En concreto, el sistema sugerirá un tipo de restaurante que el usuario podría disfrutar, orientado a personas abiertas a experimentar y descubrir nuevas opciones gastronómicas. Así, el problema se plantea en un contexto que combina datos y decisiones, proponiendo una solución dinámica e innovadora.

El problema a estudiar lo hemos nombrado: "¿Qué comemos hoy?", en este, se deben dar de entrada dos variables cuantitativas: "Edad" y "Presupuesto". Para esta último hemos decidido crear 4 clases de elección, que son las siguientes: 10 - 20 €, 20 - 30 €, 30 - 50 € y más de 50 €.
(Hemos decidido esta clasificación guiándonos en base a la clasificación de google para restaurantes).

Finalmente hay solo una variable de salida cualitativa, "Gustos" que se clasifica en cuatro tipos: Italiano, Español, Asiático y Fast food.
(Todo esto está orientado a una única persona).

# Capítulo 2

# ¿Qué datos utilizar?

Los datos que se deben utilizar para este tipo de problemas han de ser lo más representativos y realistas posibles, es decir, no resulta adecuado generarlos aleatoriamente, pues no se encontraría la relación que se desea entre edad, precios y gustos. Para mayor claridad, no se espera encontrar más de uno o dos casos, en un conjunto aproximado de 100 datos, de un joven de 20 años que se gaste más de 50 € en comer o una persona de 70 años que se gaste entre 10 y 20 € en comida tipo fast food, y esto no nos lo pueden ofrecer unos datos aleatorios. (En ningún momento se está afirmando que no sean posibles estos casos, solo que no son muy habituales.)

Con este propósito, se realizaron encuestas a más de 100 personas, recopilando información sobre su edad, el gasto promedio en comida y sus preferencias entre cuatro tipos de restaurantes. Afortunadamente, los resultados obtenidos muestran una distribución de datos coherente con el razonamiento previamente expuesto, lo que sugiere que las respuestas están adecuadamente equilibradas.

En la siguiente tabla podemos observar cómo son nuestros datos exactamente, con cinco datos representativos, en el que cada persona ha indicado su edad, su presupuesto a la hora de comer y su gusto.

| **Personas** | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|
| **Edad** | 21 | 30 | 41 | 58 | 80 |
| **Presupuesto** | 10 - 20 | 30 - 50 | 50+ | 30 - 50 | 50+ |
| **Gusto** | Fast Food | Asiático | Italiano | Español | Español |

Cuadro 1: Datos.

A continuación, se procede a la representación gráfica de los datos, comparando la variable edad con el presupuesto asignado. Para mejorar la visualización y facilitar la diferenciación de los gustos individuales, se ha utilizado un esquema de colores, asignando un color distinto a cada uno de los gustos. De esta manera, la gráfica también proporciona una clara distinción entre los diferentes gustos presentes en el conjunto de datos analizados, permitiendo observar si existe alguna relación entre estos y las variables independientes.

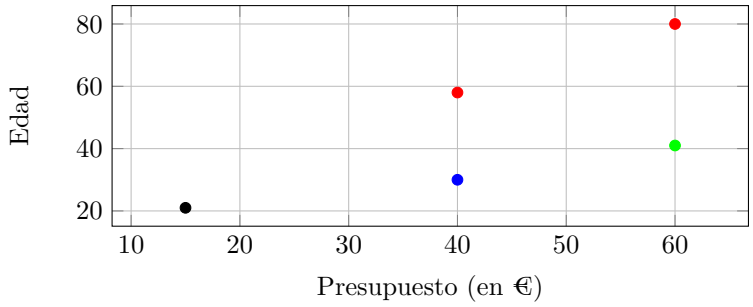


Figura 1: Gráfico de la Edad vs. Presupuesto.

# Capítulo 3

# Entender y preprocesar los datos.

Los datos utilizados en este estudio fueron generados manualmente, por lo que no fue necesario preprocesarlos. Sin embargo, se llevó a cabo una modificación en la clasificación inicial de los restaurantes: originalmente habíamos considerado cinco categorías distintas, pero se decidió reducirlas a cuatro, combinando la comida india y la oriental en una sola categoría, denominada "Asiática", para simplificar el análisis.

Además dentro de la clasificación de "presupuesto", como trabajamos con intervalos, que representarían una variable cualitativa, si la respuesta dada es 10 - 20€, se le asigna a ese elemento un número aleatorio entre 10 y 20, de forma que, entonces, sea una variable cuantitativa. Este mismo proceso se repite para 20 - 30€, 30 - 50€ y más de 50€.

Existe una razón por la que tomamos un número aleatorio para cada intervalo: si escogiéramos por ejemplo la media del intervalo, para cada dato con una misma edad y mismo intervalo repetido, podría ser que esas dos personas con, por ejemplo, 20 años, no gastaran exactamente el mismo dinero. Sin embargo, los dos puntos que los representan se solaparían en la gráfica. Como acabamos de decir, este fenómeno también afecta a la hora de representar los datos y visualizarlos, de manera que se distribuyen de una forma más uniforme en la gráfica si se les atribuye un número aleatorio y se rellenan muchos huecos vacíos, a la vez que se van a representar "casi" todos los datos y ninguno se va a solapar.

# Capítulo 4

# Elección de la metodología a estudiar.

Para poder elegir un modelo adecuado, es esencial entender la naturaleza del problema. En este caso, se trata de un problema con dos variables cuantitativas de entrada y una categórica de salida. Por este motivo, vamos a emplear un modelo logístico multinomial que permita devolver variables categóricas. Dado que buscamos dar una respuesta, debe ser un modelo supervisado, y además, priorizaremos aquellos especializados en dar recomendaciones.

Entre los modelos considerados para este estudio se encuentran el modelo Naive de Bayes [5] y el modelo de regresión logística multinomial [3] [4]. Ambos son apropiados para problemas de clasificación multiclase, sin embargo, presentan diferencias significativas que influyen en su aplicabilidad. Estas diferencias nos han llevado a la selección del modelo de regresión logística.

La diferencia principal es que el modelo Naive de Bayes asume independencia total entre las variables, mientras que el de regresión logística sí considera la posibilidad de que haya dependencia, lo cual cabe esperar que suceda, pues habrá una relación entre las variables "Edad" y "Presupuesto". Es cierto que esta suposición simplifica los cálculos, pero lleva a resultados menos precisos si las variables están altamente correlacionadas.

A favor del modelo elegido, este es mucho más interpretable ya que los coeficientes asociados a cada categoría muestran el impacto directo sobre las probabilidades y, en contra de este, su rendimiento es menor y su complejidad computacional causa que sea más costoso.
Luego, es un modelo que trabaja mejor con conjuntos de datos más pequeños.

# Capítulo 5

# Implementación práctica de la metodología en la base de datos.

Como ya hemos comentado en el apartado anterior, trabajaremos con el modelo de regresión logística multinomial, que es un método estadístico utilizado para modelar situaciones en las que la variable dependiente es categórica y tiene más de dos categorías.

## 5.1. ¿Qué hace?

Este método estima la probabilidad de que una observación pertenezca a una de las posibles categorías de la variable dependiente, basándose en una o más variables predictoras o independientes. En nuestro caso, con cuatro categorías distintas la variable dependiente, "gusto" y dos variables independientes, "edad" y "presupuesto", el modelo estima la probabilidad de que a una observación se le asigne una de esas categorías. [1]

## 5.2. Fórmula.

Para cada k, donde $k = \{1, 2, 3, \ldots, K\}$ (en nuestro ejemplo, $K = 4$), asignamos una función de probabilidad:

$$P(y = k \mid x) = \frac{e^{\beta_k x}}{\sum_{j=1}^{K} e^{\beta_j x}}$$

Donde $\beta_k$ es el vector de coeficientes correspondientes de la categoría $k$ y la suma del denominador asegura que todas las probabilidades sumen 1. [3]

**¿Por qué de esta fórmula?**

1. **Función exponencial:**
   En esta formulación, la probabilidad de un resultado $Y_i = k$ se expresa utilizando un predictor lineal y un término adicional de normalización, la función de partición $Z$, de la siguiente manera:

   $$\ln \Pr(Y_i = k) = \boldsymbol{\beta}_k \cdot \mathbf{X}_i - \ln Z, \quad k \leq K.$$

   Para asegurar que el conjunto de probabilidades forme una distribución válida (es decir, que la suma de probabilidades para todas las categorías posibles sea igual a 1), se introduce un término de normalización $-\ln Z$. Esto garantiza que:

   $$\sum_{k=1}^{K} \Pr(Y_i = k) = 1.$$

Después de exponenciar ambos lados, la probabilidad se puede escribir como:

$$\Pr(Y_i = k) = \frac{1}{Z} e^{\boldsymbol{\beta}_k \cdot \mathbf{X}_i}, \quad k \leq K,$$

donde $Z$, la función de partición, asegura la normalización y se define como:

$$Z = \sum_{k=1}^{K} e^{\boldsymbol{\beta}_k \cdot \mathbf{X}_i}.$$

Además, la función exponencial asegura que las probabilidades sean siempre positivas. [8]

2. **Término $\boldsymbol{\beta_k x}$:**
   Es una función lineal de las variables independientes que "mapea" los valores de $x$ a un valor que puede ser interpretado como una escala para cada categoría $k$. Cuanto mayor sea este valor mayor es la atracción hacia la categoría k. [4]

## 5.3. ¿Cómo elige el modelo los parámetros?

Para estimar los coeficientes utilizamos un proceso llamado máxima verosimilitud. La idea principal detrás de este método es encontrar los valores de $\beta_k$ (que son únicos para cada categoría) que maximizan la probabilidad de pertenecer a la categoría k.
La función de verosimilitud que utilizamos es: $L(\beta) = \prod_{i=1}^{N} P\left(y_i \mid x_i\right)$ donde N es el número total de observaciones.
En lugar de trabajar con esta expresión, aplicaremos el logaritmo, para operar con sumas en vez de productos.

El objetivo principal es encontrar los coeficientes $\beta$ que maximicen la función anterior. Para ello hacemos uso de técnicas de optimización numérica como el método de Newton-Raphson [6] o el gradiente decreciente.

Para explicar el método de Newton-Raphson, mostremos un ejemplo en concreto [2]. Dada una función que va de $\mathbb{R}^{12}$ a $\mathbb{R}$, consideremos $f : \mathbb{R}^{12} \to \mathbb{R}$:

$$f(x_1, x_2, \ldots, x_{12}) = x_1^2 + x_2^2 + \cdots + x_{12}^2 - 1$$

1. **Punto inicial:** escoge un punto inicial, por ejemplo, $\mathbf{x_0} = (0{,}1, 0{,}1, \ldots, 0{,}1)$.

2. **Gradiente:** calcula el gradiente $\nabla f$: [12]

$$\nabla f(x_1, x_2, \ldots, x_{12}) = \begin{pmatrix} 2x_1 \\ 2x_2 \\ \vdots \\ 2x_{12} \end{pmatrix}$$

3. **Evaluación en el punto inicial:** calcula $f(\mathbf{x_0})$ y $\nabla f(\mathbf{x_0})$.
4. **Resolver el sistema:** resuelve el sistema $\Delta x = -\frac{f(\mathbf{x_0})}{\|\nabla f(\mathbf{x_0})\|^2}\nabla f(\mathbf{x_0})$.
5. **Actualizar el punto:** actualiza $\mathbf{x}$ como $\mathbf{x_1} = \mathbf{x_0} + \Delta x$.
6. **Iterar:** repite los pasos hasta que $f(\mathbf{x})$ esté lo suficientemente cerca de 0.

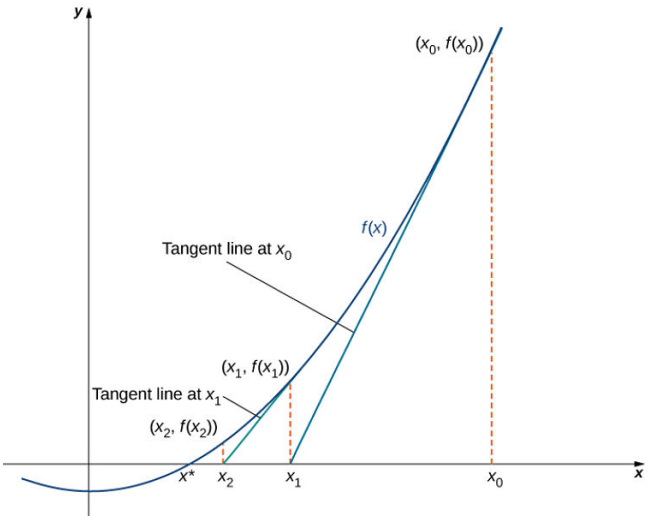


Figura 2: Ejemplo gráfico de una función en $\mathbb{R}^2$.

Para asegurar su convergencia, es necesario que se cumplan las siguientes condiciones:

1. $f$ sea **continuamente diferenciable**: la función objetivo debe ser suave, lo que garantiza la existencia y continuidad de las derivadas [9].
2. El **gradiente** $\nabla f(x)$ no sea cero en el entorno de la solución: el gradiente de la función (un vector en $\mathbb{R}^{12}$) indica la dirección de mayor cambio. Si el gradiente es cero, el método pierde información sobre hacia dónde moverse.
3. El **Hessiano** $Hess(f)(x)$ sea **invertible** y preferiblemente definido positivo: el Hessiano es la matriz de segundas derivadas. Debe ser invertible para que las actualizaciones del método sean válidas, y si es definido positivo, garantiza que se está cerca de un mínimo local.
4. El **punto inicial** esté razonablemente cerca de la solución: el método tiene convergencia local, por lo que el punto inicial debe estar lo suficientemente cerca de la solución para asegurar que converja correctamente [7].

Es fundamental tomar en cuenta tanto el gradiente como el Hessiano, dado que la función tiene muchas variables de entrada (12) y una única salida real.

## 5.4. Funcionamiento.

Vamos a hacer un ejemplo en el que calcularemos las probabilidades paso a paso con un número reducido de datos para entender cómo trabaja el modelo.

| Edad | Presupuesto | Categoría |
|---|---|---|
| 30 | 40 | 1 |
| 30 | 60 | 2 |
| 70 | 20 | 3 |
| 45 | 50 | 2 |
| 65 | 30 | 4 |

Cuadro 2: Datos de la muestra.

La fórmula a utilizar para calcular la probabilidad es:

$$P(y = k|x) = \frac{e^{\beta_k x}}{\sum_{j=1}^{4} e^{\beta_j x}} \quad \text{para } k = 1, ..., 4.$$

El siguiente paso es calcular los coeficientes $\beta_k$ para cada categoría $k = 1, ..., 4$. Observar que la expresión de $\beta_k$ es la siguiente:

$$\beta_k = (\beta_k^0, \beta_k^1, \beta_k^2)$$

Mientras que la de $x_i$ es:

$$x_i = (x_i^1, x_i^2)$$

Entonces, para tener en cuenta el intercepto y poder operar ambos vectores, necesitamos añadir una componente al $x_i$, al que denotaremos $\tilde{x}_i = (1, x_i^1, x_i^2)$. Ahora, ya podemos proceder a estimar $\beta_k$, considerando la función de máxima verosimilitud.

$$L(\beta) = \prod_{i=1}^{n} \frac{e^{\beta_{y_i} \tilde{x}_i}}{\sum_{j=1}^{4} e^{\beta_j \tilde{x}_i}}$$

Donde, $\beta_{y_i}$ es el vector de coeficientes correspondiente a la categoría observada $y_i$ por la observación $i$.
Maximizar esta expresión es complejo, es por eso que buscamos simplificar el producto de probabilidades, tomando el logaritmo de la expresión anterior.

$$log(L(\beta)) = \sum_{i=1}^{n} log\left(\frac{e^{\beta_{y_i} \tilde{x}_i}}{\sum_{j=1}^{4} e^{\beta_j \tilde{x}_i}}\right) \quad \Rightarrow \quad log(L(\beta)) = \sum_{i=1}^{n} (\beta_{y_i} \tilde{x}_i - log(\sum_{j=1}^{4} e^{\beta_j \tilde{x}_i}))$$

En este punto, maximizar esta función a mano es demasiado dificultoso, luego hacemos uso del método de optimización Newton-Raphson.

Una vez calculados, tenemos los siguientes datos:

| Categoría | Coeficiente de Edad | Coeficiente de Presupuesto | Intercepto |
|---|---|---|---|
| 1 | $\beta_1^1 = 0{,}1$ | $\beta_1^2 = 0{,}05$ | $\beta_1^0 = -1{,}5$ |
| 2 | $\beta_2^1 = 0{,}15$ | $\beta_2^2 = 0{,}04$ | $\beta_2^0 = -1{,}2$ |
| 3 | $\beta_3^1 = -0{,}05$ | $\beta_3^2 = 0{,}02$ | $\beta_3^0 = -1{,}0$ |
| 4(ref) | $\beta_4^1 = 0$ | $\beta_4^2 = 0$ | $\beta_4^0 = 0$ |

Cuadro 3: Coeficientes $\beta$.

Vamos a calcular las probabilidades para una observación concreta:

- Edad: 55.
- Presupuesto: 45.

Ahora determinamos $\beta_k \tilde{x}$ (los predictores lineales) para cada categoría, usando los coeficientes.

- Categoría 1: $\beta_1 \tilde{x} = -1{,}5 + 0{,}1 \cdot 55 + 0{,}05 \cdot 45 = 6{,}25$
- Categoría 2: $\beta_2 \tilde{x} = -1{,}2 + 0{,}15 \cdot 55 + 0{,}04 \cdot 45 = 8{,}85$
- Categoría 3: $\beta_3 \tilde{x} = -1{,}0 - 0{,}05 \cdot 55 + 0{,}02 \cdot 45 = -2{,}85$
- Categoría 4: $\beta_4 \tilde{x} = 0$

Calculamos las exponenciales:

- $e^{\beta_1 \tilde{x}} = e^{6{,}25} = 518{,}012$
- $e^{\beta_2 \tilde{x}} = e^{8{,}85} = 6978{,}665$
- $e^{\beta_3 \tilde{x}} = e^{-2{,}85} = 0{,}058$
- $e^{\beta_4 \tilde{x}} = e^0 = 1$

Obtenemos el denominador: $\sum_{j=1}^{4} e^{\beta_j \tilde{x}} = 7497{,}735$
Por último, hallamos las probabilidades:

- $P(y = 1|x) = \frac{e^{6{,}25}}{7497{,}735} = 0{,}0641$
- $P(y = 2|x) = \frac{e^{8{,}85}}{7497{,}735} = 0{,}9307$
- $P(y = 3|x) = \frac{e^{-2{,}85}}{7497{,}735} = 0{,}0000077$
- $P(y = 4|x) = \frac{e^0}{7497{,}735} = 0{,}0001334$

Resultado final: según nuestro modelo, es altamente probable que la persona escoja la categoría 2.

## 5.5. Aplicación a nuestros datos.

En este apartado, comenzaremos a trabajar con los datos obtenidos y aplicaremos los conocimientos previamente introducidos en un código en Python.

Antes de avanzar, surge la pregunta de ¿qué precisión es razonable esperar? Para modelos simples con dos variables independientes y una variable de respuesta con cuatro clases (multiclase), la precisión base dependerá, evidentemente, de la complejidad de los datos.

La precisión base, es decir, la aleatoria, sería del 25 % si las clases están distribuidas uniformemente (dado que hay cuatro posibles clases). Cualquier modelo que supere este valor estaría mejorando las predicciones en comparación con una asignación aleatoria.

Por otro lado, modelos bien ajustados podrían alcanzar precisiones significativamente mayores, situándose en un rango del 60 % al 90 % en algunos casos, dependiendo de los factores mencionados.
En la primera prueba de nuestro código, hemos decidido tomar pocos datos y observar qué ocurre, tomando de forma aleatoria el $x_{train}$ y el $x_{test}$.
En nuestra gráfica, la figura 1, tenemos una precisión del 26 %:

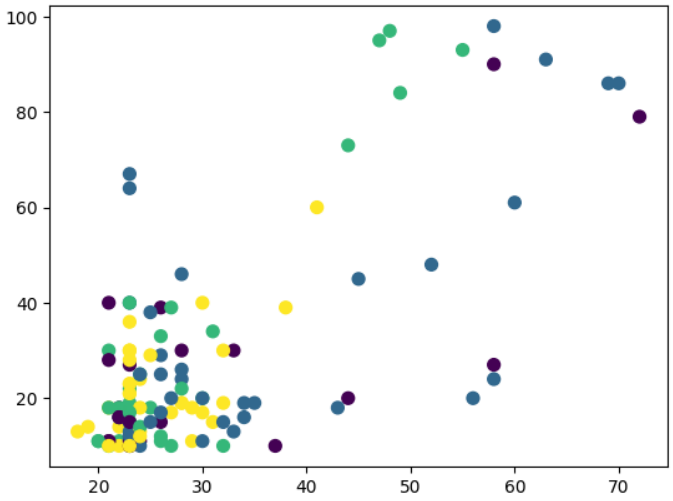


Figura 3: Precisión de 0.26.

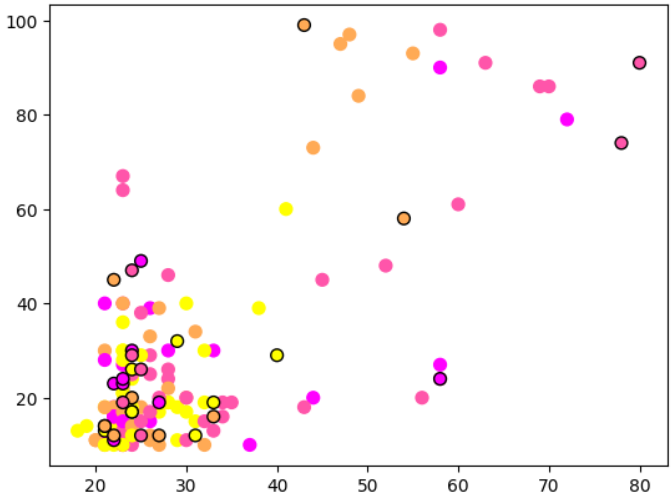


Figura 4: Mostrando los elementos test.

Como podemos observar, la precisión obtenida es representativa de unos datos aleatorios, es decir, muy baja. Esta situación puede deberse a diversas razones, tales como una mala distribución de los datos, una cantidad insuficiente de estos, una inadecuada selección de los datos de train, etc. Sin embargo, existen múltiples estrategias para abordar estas deficiencias, como realizar una encuesta que proporcione un conjunto de datos más equilibrado o aplicar técnicas de validación cruzada.

Después veremos que, efectivamente, es por la baja linealidad de los datos. Primero comprobemos que realmente nuestros datos no están adecuadamente balanceados:

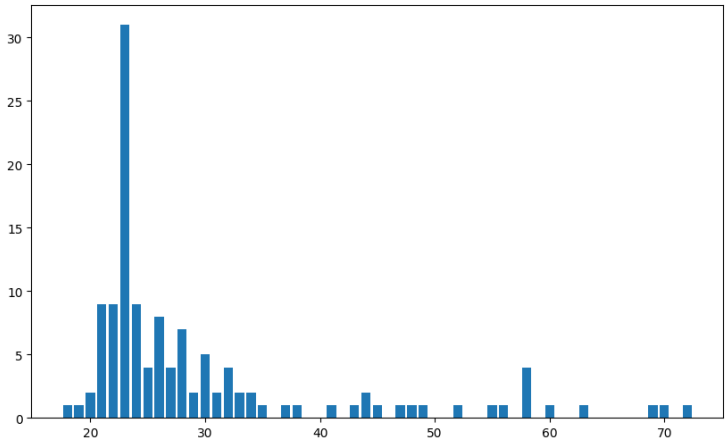


Figura 5: Diagrama de barras de los datos.

Es evidente que contamos con una sobreabundancia de datos correspondientes a individuos de entre 20 y 30 años, mientras que los grupos de edad restantes están subrepresentados, lo que sugiere una falta de variedad en nuestro conjunto de datos.

Evaluemos si la inclusión de más datos puede mejorar nuestra situación. Para ello, aplicaremos la curva de aprendizaje, que analiza cómo variará nuestro resultado al añadir más datos aleatorios que sigan la misma distribución que los datos actuales.

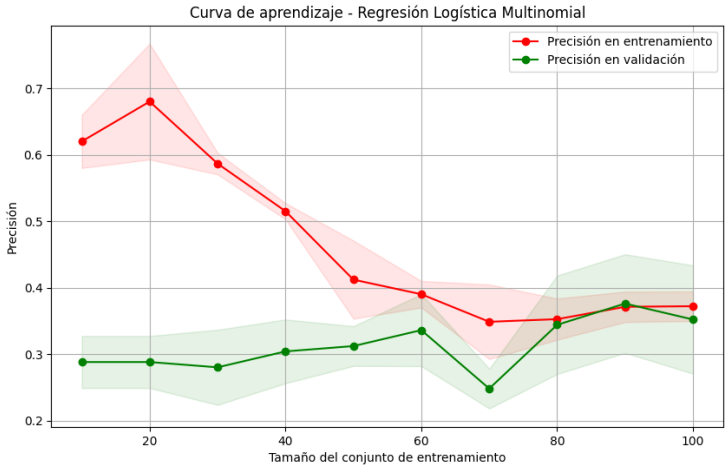


Figura 6: Curva de aprendizaje

Podemos observar que su precisión tiende aproximadamente a 0.35 a medida que aumentamos el número de datos.
Aunque es evidente que la precisión mejora con una mayor cantidad de datos, esta cifra no es suficientemente buena. Esto sugiere que el verdadero problema radica en la distribución de los datos, más que en su cantidad.
Veamos un pequeño ejemplo, si introducimos el elemento (34, 22), es decir, una persona de 34 años de edad dispuesta a gastarse 22€ en la comida, nuestro modelo devuelve: "El tipo de restaurante recomendado para una persona de 34 años dispuesta a gastarse 22 € es: español".

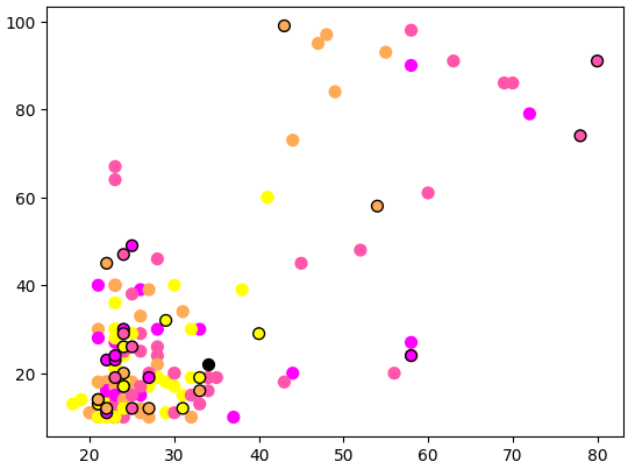


Figura 7: El punto negro es nuestro nuevo dato.

No obstante, procederemos a recrear el problema utilizando un conjunto de datos mucho más amplio para determinar si los resultados anteriores fueron simplemente una coincidencia.

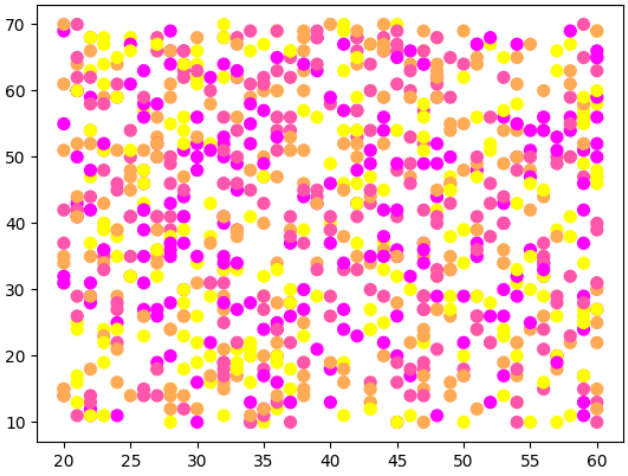


Figura 8: Obtenemos una precisión de 0.31.

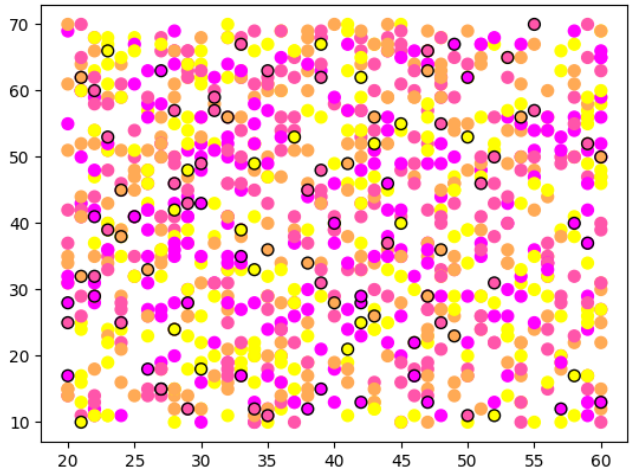


Figura 9: Mostrando los elementos test.

Como observamos nos da un resultado muy similar a las curvas de aprendizaje, así que no parece ser una coincidencia.

Verifiquemos si al modificar la distribución de nuestros datos logramos aumentar la precisión del modelo. Para ello, tomamos nuevos datos que nos proporcionen la variedad que nos faltaba, con una cantidad mayor de estos y siguiendo una distribución más equilibrada. Posteriormente, dividimos los datos en conjuntos de entrenamiento y prueba de manera aleatoria.

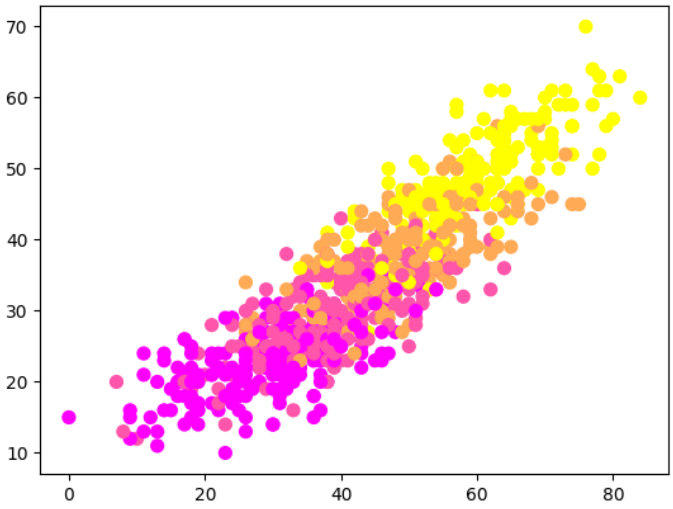


Figura 10: Obtenemos una precisión de 0.645.

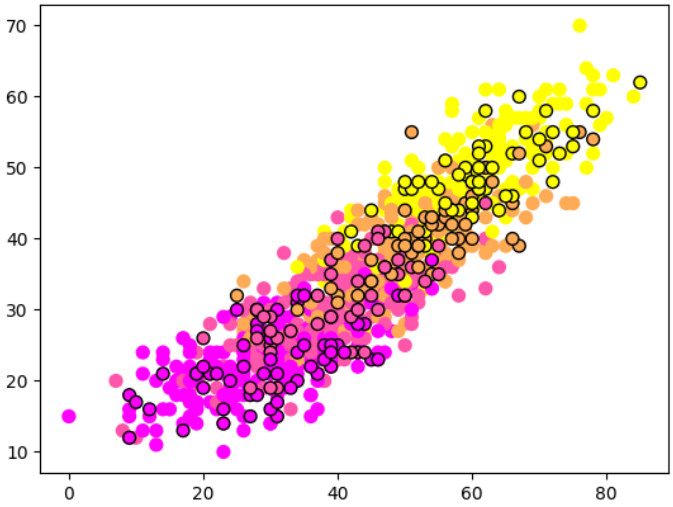


Figura 11: Mostramos los elementos test.

A nivel visual, podemos observar que los datos ahora están más equilibrados. Además, su precisión ha mejorado considerablemente, lo que nos permite afirmar que es una estimación válida para nuestros datos.
Para confirmar si esta mejora es una casualidad o si nuestro modelo realmente ha progresado con la inclusión de los nuevos datos, procederemos a analizar la curva de aprendizaje.

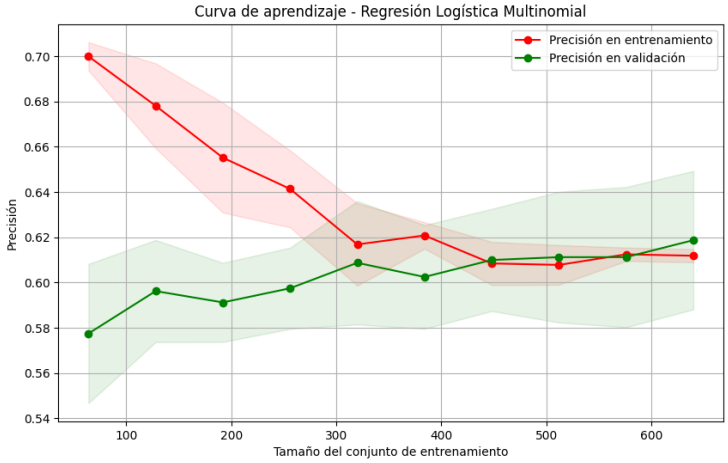


Figura 12: Podemos observar como la curva de aprendizaje tiende aproximadamente a 0.61 a medida que aumentamos los datos.

Con lo anterior, podemos afirmar que los datos son adecuados y que esta nueva distribución es efectiva.
Si ahora introducimos de nuevo el elemento (34, 22), nuestro modelo devuelve: "El tipo de restaurante recomendado para una persona de 34 años dispuesta a gastarse 22 € es: italiano".

## 5.6. Búsqueda de hiperparámetros.

Con el objetivo de mejorar la precisión de nuestro modelo, nos centramos ahora en determinar los hiperparámetros óptimos para nuestro modelo, empleando la función `GridSearchCV` de Python [11]. Esta herramienta permite explorar y representar todas las combinaciones posibles de hiperparámetros que se pueden evaluar en el modelo de machine learning seleccionado. Los parámetros de entrada de esta función incluyen, entre otros, las cuadrículas de los hiperparámetros, el modelo a optimizar y opciones adicionales como el número de pliegues utilizados en la validación cruzada.

Por ejemplo, en el caso de tener dos hiperparámetros, cada uno con dos posibles valores, el número total de combinaciones a evaluar ascendería a cuatro. Posteriormente, `GridSearchCV` aplica la técnica de validación cruzada, dividiendo los datos en $k$ pliegues. Este proceso implica entrenar el modelo $k$ veces, utilizando $k-1$ pliegues para el entrenamiento y el pliegue restante para la validación. Finalmente, se promedian los resultados obtenidos (ya sea en términos de error o precisión) para cada combinación de hiperparámetros, lo que permite identificar la configuración más efectiva para el modelo. [10]

Al final del proceso, la función se encarga de identificar la combinación de hiperparámetros que maximiza el rendimiento del modelo, evaluado según una métrica predefinida. Este enfoque tiene como objetivo mejorar la eficacia del modelo en tareas de predicción.

La implementación de la validación cruzada dentro de `GridSearchCV` juega un papel crucial en la mitigación del sobreajuste, permitiendo así una mejor generalización del modelo. Esto se traduce en un aumento de la robustez del mismo, lo cual es fundamental en la práctica del machine learning.

Es relevante destacar que `GridSearchCV` no se fundamenta en un único modelo teórico; en cambio, integra diversos procesos estadísticos, técnicas de validación cruzada y métodos de optimización.

Aplicamos esta metodología al caso particular bajo estudio, con el objetivo de optimizar dos parámetros clave utilizados por la función `LogisticRegression` en Python. Estos parámetros son la tolerancia ($tol$) y una constante de regularización ($C$), la cual desempeña un papel crucial en el cálculo de los coeficientes del modelo. Tras un análisis exhaustivo, determinamos que la combinación óptima de parámetros es $C = 100$ y $tol = 0{,}0001$, logrando una precisión del 68 %.

## 5.7. Introducción de una nueva variable.

Durante la ronda de preguntas en la presentación intermedia, surgió la duda acerca de si la precisión de nuestro modelo mejoraba significativamente al introducir una nueva variable independiente. Por lo tanto, decidimos investigar brevemente esta cuestión.

Generamos nuevos datos mediante inteligencia artificial, manteniendo una distribución similar entre las dos variables independientes previamente utilizadas, pero incorporando una nueva variable que especifica el sexo de los individuos, codificado como 0 = "hombre" y 1 = "mujer".

Implementamos nuestro modelo a estos datos y vemos como trabaja:

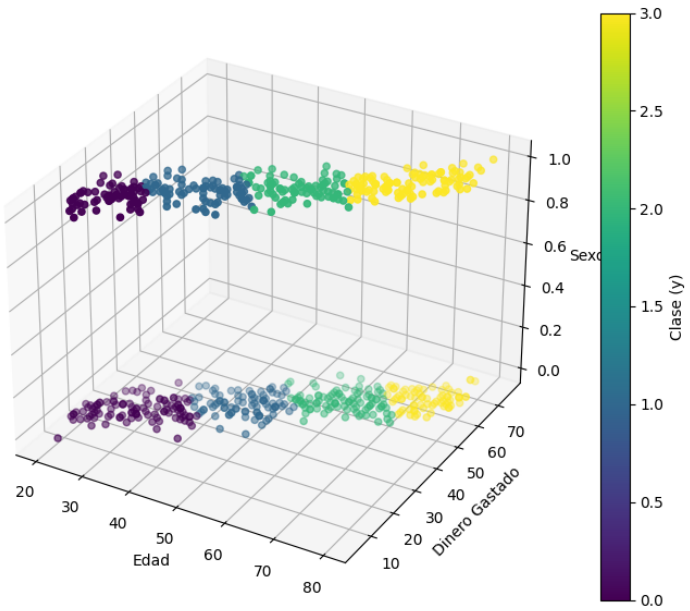


Figura 13: Representación de los datos tras la inclusión de la tercera variable.

Realizando un estudio de la precisión observamos que esta mejora notablemente, alcanzando un 94 %, lo que representa un incremento considerable en el rendimiento. Sin embargo, en este simple estudio nos hemos percatado que también aumenta ligeramente el tiempo de compilación del código.

En conclusión, la inclusión de una nueva variable, basada en datos bien distribuidos, proporciona una estimación mucho más precisa. Esto subraya la relevancia de ajustar adecuadamente las variables de entrada, considerando la distribución de la variable categórica.

# Capítulo 6

# Conclusiones y posibles mejoras.

En este estudio, hemos desarrollado un modelo de regresión logística multinomial para predecir las preferencias alimenticias de los individuos en función de su edad y presupuesto. Los resultados muestran que el modelo puede proporcionar recomendaciones de restaurantes con una precisión razonable (64.5 %) cuando se dispone de un conjunto de datos equilibrado. A pesar de los logros alcanzados, el modelo todavía presenta limitaciones en su capacidad para generalizar a nuevas poblaciones debido a la falta de diversidad en los datos.

Estas son algunas de las posibles medidas que podríamos tomar para mejorar nuestro modelo:

- **Diversificación del conjunto de datos:** realizar más encuestas a personas de diferentes rangos etarios y presupuestarios para asegurar una mejor representación y, en consecuencia, mejorar la robustez y precisión del modelo.
- **Nuevas variables:** incluir factores adicionales como restricciones dietéticas, género (esta se añadió en un breve ejemplo, véase en 5.7) o nacionalidad, que pueden influir en las preferencias alimenticias, y, por tanto, podrían conducir a resultados aún más precisos y personalizados.
- **Validación cruzada avanzada:** implementar técnicas adicionales de validación cruzada para mitigar posibles sobreajustes y mejorar la generalización del modelo (se empleó una búsqueda de hiperparámetros, véase en 5.6).

Otras medidas que podríamos implementar con la intención de aumentar la utilidad de nuestro modelo son:

- **Aumento de categorías de salida:** disponer de más clases de comida, de esta manera, la personalización se vería incrementada y, en consecuencia, la precisión también sería mayor.
- **Repetición:** se podría aplicar el modelo una segunda vez, una vez se ha obtenido una respuesta y se le ha asignado un tipo de comida al usuario, según su presupuesto, dicho gusto y la ciudad que habita, se devolvería un restaurante de acorde a estas preferencias.

# Capítulo 7

# Reflexión del trabajo en grupo.

En este proyecto, hemos logrado una comprensión profunda del proceso de desarrollo de modelos de machine learning y de cómo los datos influyen directamente en la calidad de las predicciones.
Respecto al trabajo en grupo, no hemos enfrentado ninguna dificultad, es más, nos hemos complementado de manera efectiva. Antes de empezar, discutimos acerca de los puntos fuertes y débiles de cada integrante del grupo, de forma que pudiéramos organizar las tareas de acuerdo con esas consideraciones. La idea era potenciar nuestras habilidades, a la vez que aprendíamos de las de los demás.

La dinámica grupal ha resultado ser exitosa: antes de comenzar a trabajar, decidíamos qué tareas debía hacer cada miembro del grupo, en algunas ocasiones dividiéndonos en grupos de dos, y en otras, individualmente. Posteriormente, cada uno compartía su progreso y las dificultades que habían surgido. Cada miembro ha dado su opinión en cada aspecto que ha considerado necesario, y siempre se han tenido en cuenta todas las perspectivas.

Especifiquemos un poco sobre el proceso grupal del trabajo:

En las etapas iniciales, realizamos una sesión de lluvia de ideas para definir la temática del proyecto, y finalmente surgió la propuesta de desarrollar un modelo de recomendación de restaurantes basado en los gustos de los usuarios. Durante el proceso de toma de decisiones, evaluamos varias opciones para elegir la metodología a utilizar (¿qué tipo de modelo en específico?, ¿aprendizaje supervisado o no supervisado?...), decantándonos por un enfoque supervisado y el modelo de regresión logística multinomial. Se llevaron a cabo encuestas en Instagram para la recopilación de datos, mientras que, simultáneamente, se avanzaba en el desarrollo del código.

Tras recopilar algo de información, se inició la creación de la presentación en Beamer, organizando las secciones y seleccionando una plantilla adecuada. Además, se empezó a elaborar el trabajo escrito, que serviría también como guion inicial para la exposición. También se continuó refinando el modelo informático, incorporando gráficas y nuevos enfoques, como el análisis de precisión en función de los datos. Es importante destacar que al final de cada sesión, se compartían los avances y se organizaban en función de las tareas de cada miembro del grupo.

Tras la presentación, nos enfocamos en varios aspectos, destacando como prioritario el desarrollo de un ejemplo de regresión logística multinomial utilizando un conjunto de datos reducido y sencillo.

Finalmente, en la etapa de conclusión del proyecto, se terminó la redacción de este documento, añadiendo contenidos que cada uno consideraba relevantes. Además, cualquier sugerencia o modificación propuesta por los integrantes era consultada y consensuada por el grupo antes de ser implementada.

Durante el proyecto han aparecido algunas complicaciones, como pueden ser:

- La poca precisión que nos ofrecía el modelo en una primera instancia (véase en 5.5).
- La implementación de un ejemplo simple teórico del modelo de regresión logística multinomial, con la búsqueda de los coeficientes de cada clase, y el entendimiento de procesos de optimización (véase en 5.4).
- La búsqueda de hiperparámetros y entender cómo funciona `GridSearchCV` (véase en 5.6).

Después de haber terminado este proyecto, no solo hemos aprendido sobre modelos de inteligencia artificial, concretamente sobre el de regresión logística multinomial, sino que también hemos mejorado nuestra capacidad de trabajo en equipo, sabiendo delegar tareas en los demás y aprendiendo de las fortalezas del resto. Por último, los roles no han supuesto ningún problema, ya que todos eramos capaces de estar en cualquiera de ellos.

# Bibliografía


[1] Tamara Broderick. Mit: Machine learning 6.036, lecture 4: Logistic regression (fall 2020), 2020.

[2] R. L. Burden and J. D. Faires. *Numerical Analysis*. Brooks/Cole, Boston, MA, 9th edition, 2011.

[3] Wikipedia contributors. Logistic regression, 2024.

[4] Wikipedia contributors. Multinomial logistic regression, 2024.

[5] Wikipedia contributors. Naive bayes classifier, 2024.

[6] Wikipedia contributors. Newton's method, 2024.

[7] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, PA, 1996.

[8] MIT Open Learning Library. Logistic regression - mitx 6.036, 2019.

[9] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, NY, 2nd edition, 2006.

[10] scikit-learn developers. Cross-validation: evaluating estimator performance, 2024.

[11] scikit-learn developers. sklearn.model_selection.gridsearchcv, 2024.

[12] G. Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, Wellesley, MA, 5th edition, 2016.