

Notas de Clase:

Aplicaciones de Machine Learning a través de R y Python, y Algunos Temas Selectos para el Análisis de Datos

Benjamín Oliva ¹

Draft Septiembre 2024

¹benjov@ciencias.unam.mx y <https://github.com/benjov>

Documento siempre proceso de mejora.
Comentarios, dudas, etc., siempre serán
bienvenidos...

Índice general

1. Aplicaciones de Machine Learning a través de R y Python	3
1.1. Introducción: El concepto de esperanza condicional y Causalidad	3
1.1.1. Notación en el análisis de regresión	5
1.1.2. Introducción a inferencia causal	7
1.2. 4.1 Aplicación de modelos de regresión lineal.	11
1.2.1. El estimador de Mínimos Cuadrados Ordinarios (MCO) y el análisis clásico de regresión	11
1.2.2. Modelos no lineales	24
1.2.3. Introducción al Aprendizaje Estadístico	29
1.2.4. 4.1.1 Regresión lineal múltiple con datos de publicidad en medios de comunicación.	32
1.2.5. Método de regresiones restringidas (Shrinkage methods). .	32
1.2.6. 4.1.2 Regresión lineal con métodos de restricción (Sh- rinkage Methods) y regularización de los parámetros. .	35
1.3. 4.2 Aplicación de modelos de regresión de respuesta binaria. .	35
1.3.1. Métodos de estimación basados en verosimilitud	35
1.3.2. Estimación de modelos no lineales	38
1.3.3. Modelos de respuesta multinomial y ordenada	42
1.3.4. 4.2.2 Regresión logística para el estudio de equidad de violencia.	49
1.3.5. 4.2.1 Regresión logística con información de videojuegos. .	49
1.4. 4.3 Aplicación de modelos clasificación mediante clustering. .	57
1.4.1. Usos de Aprendizaje No Supervisado	57
1.4.2. Análisis de componentes principales	59
1.4.3. K-Means Clustering	61
1.4.4. 4.3.1 Método de K-means con datos de epidemiológicos. .	65
1.4.5. Agrupamiento de datos de series temporales	65

1.4.6. 4.3.2 Método de K-means para series de tiempo.	65
1.5. 4.4 Aplicación de modelos de análisis de texto.	65
1.5.1. Elementos de Procesamiento de Lenguaje y Minado de Texto	65
1.5.2. 4.4.1 Ejemplos y aplicaciones de minado de texto.	87
1.5.3. 4.4.2 Implementación de modelo de lenguaje natural con N-Gramas.	87
1.5.4. 4.4.3 Clasificación de texto con Naive Bayes.	88
1.5.5. 4.4.3 Clasificación de texto con regresión logística.	88
2. Temas Selectos para el Análisis de Datos	89
2.1. 6.1 Modelos de árboles de decisión y bosques aleatorios.	89
2.2. 6.2 Modelos de redes neuronales.	92
2.2.1. Introducción y Motivación	92
2.2.2. Algunos conceptos relevantes	92
2.2.3. Preprocesamiento de datos	96
2.2.4. Entrenamiento del modelo	98
2.2.5. Ejemplo de Redes Neuronales	98
2.3. 6.3 Aplicaciones de modelos de lenguaje natural con bases vectoriales (word embeddings).	98
2.4. 6.4 Introducción a los LMMs y aplicaciones basadas en LLMs (Prompt Engineering).	99
2.4.1. Guideline for Prompt engineering	99
2.4.2. Diseño de prompts	107
Bibliografía	111

Índice de figuras

1.1.	Ejemplo 1 de un DAG	7
1.2.	Ejemplo 2 de un DAG, confusor	8
1.3.	Ejemplo 4 de un DAG, IV estimator	8
1.4.	Ejemplo 3 de un DAG, ganancias de la educación	9
1.5.	Relación entre belleza y talento, retomado de [Cun21].	11
1.6.	Ilustración de la poroyección de y en x_1 y x_2 y el significado del error e , retomado de Greene (2012, pp. 20) [Gre12]	12
1.7.	Ilustración del hiperplano generado dee regresar X_1 y X_2 en Y , retomado de Hastie, Tibshirani y Friedman (2009, pp. 45) [HTF17]	12
1.8.	Ilustración de la poroyección de y en x_1 y x_2 y el significado del error e , retomado de Hastie, Tibshirani y Friedman (2009, pp. 46) [HTF17]	13
1.9.	Funciones exponenciales, Retomado de Larsen y Marx (2012, p. 532) [LM12]	25
1.10.	Funciones logarítmicas, Retomado de Larsen y Marx (2012, p. 536) [LM12]	26
1.11.	Funciones logísticas, Retomado de Larsen y Marx (2012, p. 540) [LM12]	27
1.12.	Funciones polinomiales, retomado de https://discdown.org/flexregression/smoothreg.html	29
1.13.	División del conjunto de datos, retomado de Hastie, Tibshirani, y Friedman (2017, p. 222) [HTF17]	32
1.14.	División del conjunto de datos, retomado de Hastie, Tibshirani, y Friedman (2017, p. 222) [HTF17]	48
1.15.	Probabilidades en un modelo de respuesta ordenada, considerando $J = 4$ (retomado de Greene (2012, 788) [Gre12])	52
1.16.	En la tabla de datos ubicamos 417 tags distintas	56

1.17. US Arrest Hierarchical Clustering, retomado de Chang (2020, p 216) [Cha20]	62
1.18. Agglomerative Clustering Algorithm and Dendrogram, retomado de Chang (2020, p 219) [Cha20]	64
1.19. Some simple regex searches, retomado de Jurafsky y Martin (2023, 3)	67
1.20. The use of the brackets [] to specify a disjunction of characters, retomado de Jurafsky y Martin (2023, 3)	67
1.21. The use of the brackets [] plus the dash - to specify a range, retomado de Jurafsky y Martin (2023, 3)	68
1.22. The caret ^ for negation or just to mean ^. See below re: the backslash for escaping the period, retomado de Jurafsky y Martin (2023, 3)	68
1.23. Ejemplo de Tokenización con el modelo de OpenAI https://platform.openai.com/tokenizer	69
1.24. Retomado de Jurafsky y Martin (2023)	73
1.25. La función sigmoide $\sigma(z) = \frac{1}{1+e^{-z}}$ toma un valor real y lo mapea al rango (0, 1). Es casi lineal alrededor de 0, pero los valores atípicos se comprimen hacia 0 o 1. Retomado de Jurafsky y Martin (2023)	81
1.26. Un documento de prueba mini muestra que exhibe las características extraídas en el vector x . Retomado de Jurafsky y Martin (2023)	83
 2.1. Ejemplo de regiones. Retomado de Hastie et al (2017, 305)	90
2.2. Ejemplo de árbol. Retomado de Hastie et al (2017, 304)	91
2.3. Ejemplo de redes neuronales. Retomado de Shmueli et al (2019, 285) [Shm+19]	93
2.4. Ejemplo de la función de activación Sigmoide. Retomado de Jurafsky, et al (2024, Cap. 7, p.2) [JM23]	95
2.5. Ejemplo de la función de activación $tanh$. Retomado de Jurafsky, et al (2024, Cap. 7, p.4) [JM23]	95
2.6. Ilustración de una unidad neuronal. Retomado de Jurafsky, et al (2024, Cap. 7, p.3) [JM23]	97
2.7. Ilustración de una red neuronal con 1 capa oculta. Retomado de Jurafsky, et al (2024, Cap. 7, p.1) [JM23]	97
2.8. Retomado de https://learn.deeplearning.ai/courses/vector-databases-embedding/1/introduction	100

2.9. Retomado de https://learndeeplearning.ai/courses/vector-databases-embeddings-a-lesson/1/introduction	100
2.10. Retomado de https://learndeeplearning.ai/courses/vector-databases-embeddings-a-lesson/1/introduction	101
2.11. Retomado de Jurafsky, et al (2024, Cap. 6, p.6) [JM23]	101
2.12. Retomado de Jurafsky, et al (2024, Cap. 6, p.8) [JM23]	102
2.13. Similaridad de Coseno	102
2.14. Retomado de Jurafsky, et al (2024, Cap. 7, p.13) [JM23]	103
2.15. Retomado de Jurafsky, et al (2024, Cap. 7, p.23) [JM23]	104
2.16. Retomado de Jurafsky, et al (2024, Cap. 7, p.24) [JM23]	105
2.17. Sentiment	106
2.18. Entity	107
2.19. Information	108
2.20. LLM	109

X

Índice de cuadros

1.1. Matriz de Confusión	49
------------------------------------	----

Introducción, motivación y alcance del documento

Queremos discutir los documentos de la bibliografía como: Abadie (2021) [Aba21], Adams (2020) [Ada20], Cameron y Trivedi (2005) [CT05]; Cunningham (2021) [Cun21]; Greene (2012) [Gre12], Hastie, Tibshirani y Friedman (2017) [HTF17], James, Witten, Hastie y Tibshirani (2013) [Jam+13], Jurafsky (2023) [JM23] y Wooldridge (2010) [Woo10]; entre muchos más que utilicemos en el curso.

Conocimientos previos

Módulos I, II y III del Diplomado.

Recursos en línea y otros materiales

Direcciones del GitHub, recomendaciones de cursos en línea, materiales recomendados y demás autoaprendizaje.

1

Aplicaciones de Machine Learning a través de R y Python

1.1. Introducción: El concepto de esperanza condicional y Causalidad

El objetivo del análisis empírico en las ciencias sociales es determinar cuál es el efecto o cambio que una variable causa en otra. Por ejemplo, cuando pensamos en qué factores o variables determinan el nivel salarial promedio de las personas es posible que relacionemos el efecto que tienen los años de estudio y en particular el efecto que tiene un año adicional de estudio. Otro ejemplo puede presentarse cuando tratamos de entender el efecto que tienen las horas de estudio en las calificaciones finales de un grupo de estudiantes.

Dentro del análisis empírico se suele hacer uso de dos términos o conceptos:

1. *Ceteris paribus*
2. *Causalidad*

El primero se emplea en el análisis empírico para establecer que el efecto de una variable se sostiene siempre que asumamos que el resto de variables que pueden afectar a dicha variable que queremos explicar permanecen constantes. El segundo sirve para entender que en el análisis empírico la existencia de correlaciones no significa la presencia de causalidad. La causalidad,

en un primer momento, es la condición observada entre dos variables y deriva de la construcción teórica, modelación o racionalización que hagamos de los fenómenos que queremos analizar. De forma más sofisticada, es una relación analizada bajo un enfoque conocido como *inferencia causal*.

Partamos de lo que en estadística se conoce como **esperanza condicional**. Supongamos dos variables, y y x , que tienen una distribución conjunta y para las cuales queremos estimar el efecto que tiene la segunda en la primera. Así, es posible que utilicemos una expresión de la esperanza condicional de y en x , misma que solemos representar como:

$$\mathbb{E}[y|x, \mathbf{C}] \quad (1.1)$$

Donde \mathbf{C} representa un conjunto o vector de variables de control, en el sentido de que también explican la variabilidad de y por lo que no podemos omitirlas en un proceso de estimación de la esperanza condicional.

Dicho lo anterior, el análisis empírico que proponemos en este curso está basado en identificar correlaciones que pueden ser interpretadas como causalidad y que consiste en estimar a la ecuación (1.1). En la ecuación (1.1) hemos asumido que \mathbf{C} contiene toda la información disponible que sirve de control, por lo que es un vector que contiene sólo información que es observable. Sin embargo, no siempre es posible observar toda la información que pudiera servir de control.

Por ejemplo, al responder qué factores determinan el salario de las personas podríamos pensar en múltiples factores cuantificables y observables como: edad, sexo, años de educación, etc. No obstante, también consideraríamos factores como las habilidades propias de cada individuo, las cuales no son necesariamente observables.

Existen tres casos de los tipos de datos que son comúnmente analizados en el análisis de regresión lineal. El primero es un análisis de datos de sección cruzada en el que las ecuaciones representativas de la ecuación (1.1) son como la siguiente:

$$\ln(\text{Salario}_i) = \beta_0 + \beta_1 \text{Educacion}_i + \beta_2 \text{Experiencia}_i$$

Donde las variables de Salario_i , Educacion_i y Experiencia_i son observadas para cada uno de los individuos en la muestra es indexado por $i = 1, 2, 3, \dots, N$.

El segundo ejemplo se trata de datos de series de tiempo como la siguiente expresión:

$$\ln(\text{PIB}_t) = \beta_0 + \beta_1 \text{Empleo}_t + \beta_2 \text{Salarios}_t$$

Donde las variables PIB_t , $Empleo_t$ y $Salarios_t$ son observadas para un individuo o entidad a lo largo de una muestra del tiempo indexada por $t = 1, 2, 3, \dots, T$.

El tercer ejemplo resulta de la combinación de los dos anteriores. De esta forma podríamos observar a una muestra de un conjunto de individuos, $i = 1, 2, 3, \dots, N$, a lo largo del tiempo, $t = 1, 2, 3, \dots, T$, y analizar el comportamiento de una variable através de una ecuación como la siguiente:

$$\ln(Salario_{it}) = \beta_0 + \beta_1 Educacion_{it} + \beta_2 Experiencia_{it}$$

Donde las variables $Salario_{it}$, $Educacion_{it}$ y $Experiencia_{it}$ se observan para los mismos individuos a lo largo del tiempo que se indexan con los pares (i, t) , $i = 1, 2, 3, \dots, N$ y $t = 1, 2, 3, \dots, T$.

1.1.1. Notación en el análisis de regresión

En primer lugar estableceremos un poco de notación. En análisis de regresión siempre partimos de una representación de una ecuación lineal como:

$$y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \dots + \beta_K x_{iK} + \varepsilon_i \quad (1.2)$$

Donde cada una de las variables y_i , x_{ik} y ε_i , se observan para $i = 1, 2, 3, \dots, N$ y $k = 1, 2, \dots, K$, por lo que podremos utilizar una representación para cada i de la forma:

$$\begin{aligned} y_i &= \begin{pmatrix} x_{i1} & x_{i2} & \dots & x_{iK} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_K \end{pmatrix} + \varepsilon_i \\ &= \mathbf{X}'_i \boldsymbol{\beta} + \varepsilon_i \end{aligned} \quad (1.3)$$

La ecuación (1.3) se puede generalizar para el total de elementos en la muestra como:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad (1.4)$$

Donde:

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{pmatrix}$$

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1K} \\ x_{21} & x_{22} & \dots & x_{2K} \\ x_{31} & x_{32} & \dots & x_{3K} \\ \vdots & \vdots & \dots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{NK} \end{bmatrix}$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_K \end{pmatrix}$$

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_N \end{pmatrix}$$

En la mayoría de los casos analizados vamos a asumir que la ecuación (1.2) tiene un término constante, por lo que $x_{i1} = 1$ para todo $i = 1, 2, \dots, N$. Bajo este escenario, representaremos la matriz \mathbf{X} con una columna compuesta del número 1 (uno) en todas sus entradas, tal que, el paraméetro β_1 es un término constante en las ecuaciones (1.3) y (1.4). De esta forma la matriz anteriormente mostrada se puede ver como:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{12} & \dots & x_{1K} \\ 1 & x_{22} & \dots & x_{2K} \\ 1 & x_{32} & \dots & x_{3K} \\ \vdots & \vdots & \dots & \vdots \\ 1 & x_{n2} & \dots & x_{nK} \end{bmatrix}$$

De forma similar al caso de una regresión lineal que incluye un término constante, podemos construir matrices \mathbf{X} para casos en que alguna de las variables explicativas es dicotómica –que toma valores de 0 y 1– y casos en los que las variables explicativas han sido interactuadas –multiplicada entre sí– o transformadas mediante potencias o logaritmos.

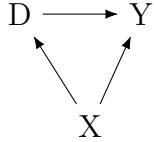


Figura 1.1: Ejemplo 1 de un DAG

1.1.2. Introducción a inferencia causal

Introducción a los Gráficos Acíclicos Dirigidos o Directed Acyclic Graphical

La notación de los Gráficos Acíclicos Dirigidos (o DAG, en inglés) representa que la causalidad va en una dirección. Así, para mostrar causalidad en reversa es necesario crear múltiples nodos. La similitud, tal como en el caso de las curvas de oferta y demanda, no tiene una representación directa o inmediata con los DAG. Otra forma plantear a los DAG es que se trata de una explicación de un fenómeno en términos de contrafactuals.

Definición 1.1 *Un DAG es una representación gráfica de una cadena de efectos causales. Los efectos causales están en sí mismos basados en algún proceso subyacente no observable.*

Los efectos causales pueden observarse en dos vías:

1. Directa: $D \rightarrow Y$
2. Indirecta, a través de una tercera variable: $D \rightarrow X \rightarrow Y$

En este sentido, la ausencia de \rightarrow indica que no hay relación entre las variables.

Un ejemplo sencillo de DAG es el siguiente:

El DAG mostrado en la Figura (1.1) ilustra que hay una ruta directa de D a Y , lo cual representa un efecto causal. Por su parte la ruta a D tiene una ruta trasera (backdoor path) a través de la ruta $D \leftarrow X \rightarrow Y$. En este caso consideraremos que la ruta directa es un efecto causal, pero la backdoor path es no causal. En su caso este proceso crea una correlación espúrea entre D y Y .

Pensemos el proceso de ruta trasera como una situación en la que a veces cuando D toma diferentes valores, Y toma diferentes valores debido a que X

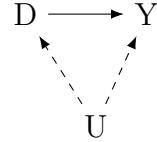


Figura 1.2: Ejemplo 2 de un DAG, confusor

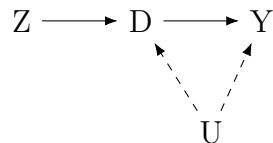


Figura 1.3: Ejemplo 4 de un DAG, IV estimator

toma diferentes valores. De esta forma, decimos que existe una correlación espuria entre D y Y .

Un segundo ejemplo de una DAG que nos permite ilustrar el concepto de confusor, el cual describe a una variable que no es observable, U , y que representamos su relación en el DAG con líneas punteadas de la forma que se muestra en la Figura (1.2).

De forma similar al caso anterior, existen dos formas para ir de D a Y . Existe la ruta directa de D a Y , lo cual representa un efecto causal. Por su parte la ruta a D tiene una ruta trasera a través de la ruta $D \leftarrow U \rightarrow Y$, pero con la diferencia de que la variable U es no observable.

¿Cómo solucionar o mitigar este efecto confusor? Una manera es conocida como usar variables instrumentales. La Figura 1.3 ilustra que U son un conjunto de factores no observables que impiden indentificar la causalidad entre D y Y . De esta forma, bucamos un Z que no está correlacionado con U pero si con D con el objeto de hace una estimación auxiliar.

Veamos otro ejemplo. Una pregunta clásica en economía y, a veces, en el discurso común es que la educación, en general, tiene el potencial de incrementar los ingresos laborales de las personas. Una posible teoría indicaría que la educación incrementa la productividad de las personas, de esta forma los trabajadores más educados deberían ser mejor pagados, ya que sus salarios son establecidos en función de su productividad. Así, la teoría afirma que la educación incrementa los ingresos.

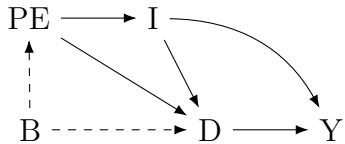


Figura 1.4: Ejemplo 3 de un DAG, ganancias de la educación

Particularicemos el ejemplo a un caso en el cual analizamos el efecto que tiene la educación secundaria (a veces denominada, educación media superior) en los ingresos. Partamos del hecho de que educarse a un nivel secundario no es un fenómeno aleatorio, puesto que existen múltiples factores que lo pueden explicar. Partamos de el siguiente DAG en el cual D será la variable de tratamiento o la variable que indica si una persona tiene formación de educación secundaria; Y es nuestra variable de interés o los ingresos; PE es la variable que indica el nivel educativo de los padres; I es una variable de ingreso familiar, y B son el background no observable de la persona (factores como la genética, habilidades individuales, ambiente familiar, etc.).

El DAG de la Figura 1.4 es en sí mismo una historia. Cada persona tiene un cierto background que normalmente no se reporta en los conjuntos de datos. Medidas como el grado de inteligencia, la personalidad, la estabilidad emocional, capacidad de resiliencia, dinámica familiar y cualquier otro relacionado con factores ambientales o del entorno no se encuentran en los datos comúnmente. Por lo que se les denomina factores no observables.

Estos factores “ambientales” (B) están correlacionados o expresados en variables de padres e hijos. Este background causa que los padres elijan un nivel educativo o ruta de educación para los hijos y también afecta las decisiones individuales de los niños respecto de la ruta educativa que quieren seguir.

En este caso, tabién es posible notar que los DAG cuentan 2 historias. Nos dicen lo que está pasando, pero tambén nos dicen lo que no está pasando. Así, B no tiene un efecto directo en los ingresos laborales, excepto a través de su efecto en la elección educativa. Sin embargo, en muchas ocasiones se suele criticar este tipo de planteamientos por parecer más un supuesto que un hecho real. En estos casos, la desición es del investigador.

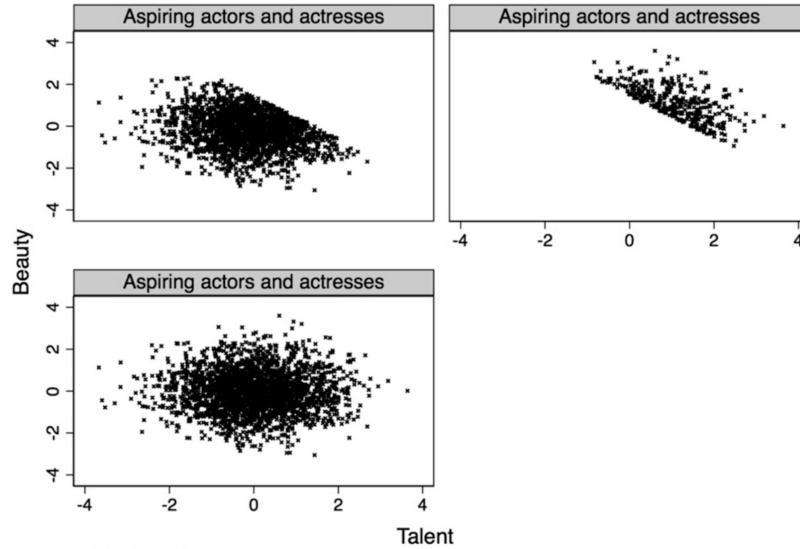
Truncamiento o cómo evitar sesgar nuestros análisis

Tomemos un ejemplo (retomado de Cunningham (2021) [Cun21]). Una publicación del blog de CNN de 2009 informó que Megan Fox, quien protagonizó la película Transformers, fue votada como la peor y la más atractiva actriz de 2009 en una encuesta sobre estrellas de cine. La publicación infería que había una implicación de que el talento y la belleza están negativamente correlacionados. ¿Pero lo son? ¿Y por qué podrían serlo? ¿Qué pasa si en realidad son independientes entre sí pero están correlacionados negativamente en una muestra de estrellas de cine debido al sesgo de cómo se seleccionó la información? ¿Es eso es posible? Es decir, ¿el talento y la belleza están negativamente correlacionados?

En la figura 1.5 muestra una simulación de la situación. El panel inferior izquierdo muestra el diagrama de dispersión entre talento y belleza. Observe que las dos variables son extracciones aleatorias e independientes de la distribución normal estándar, lo que crea una nube de datos alargada. Pero debido a que “estrella de cine” se encuentra en el percentil 85 superior de la distribución de una combinación lineal de talento y belleza, la muestra consta de personas cuya puntuación combinada se encuentra en la parte superior derecha de la distribución conjunta. Esta frontera tiene una pendiente negativa y se encuentra en la parte superior derecha de la nube de datos, lo que crea una correlación negativa entre las observaciones en la muestra de estrellas de cine. Sin embargo, sabemos que, de hecho, no hay relación entre las dos variables. Este tipo de selección de muestras crea correlaciones espurias. Una muestra aleatoria de la población completa sería suficiente para mostrar que no existe relación entre las dos variables, pero al dividir la muestra en estrellas de cine únicamente, introducimos correlaciones espurias entre las dos variables de interés.

El mensaje final es que debemos tener cuidado de no estar analizando una muestra trunca de la información y, más aún, pretender hacer inferencias generales con muestras truncas de la población.

Figura 1.5: Relación entre belleza y talento, retomado de [Cun21].



1.2. 4.1 Aplicación de modelos de regresión lineal.

1.2.1. El estimador de Mínimos Cuadrados Ordinarios (MCO) y el análisis clásico de regresión

El concepto de regresión se puede ilustrar en las Figuras 1.6, 1.7 y 1.8.

Si partimos de las ecuaciones (1.3) u (1.4), podemos establecer que el término de error estará dado por:

$$\varepsilon_i = y_i - \mathbf{X}'_i \boldsymbol{\beta} \quad (1.5)$$

Donde $i = 1, 2, \dots, N$. De forma similar podremos decir que un estimador de éste término de error será aquel que resulte de:

$$e_i = y_i - \mathbf{X}'_i \hat{\boldsymbol{\beta}} \quad (1.6)$$

Donde $\hat{\boldsymbol{\beta}}$ es un vector de estimadores de los parámetros $\boldsymbol{\beta}$. De lo dicho hasta ahora es fácil ver que la siguiente ecuación es cierta $\forall i$:

$$y_i = \mathbf{X}'_i \boldsymbol{\beta} + \varepsilon_i = \mathbf{X}'_i \hat{\boldsymbol{\beta}} + e_i \quad (1.7)$$

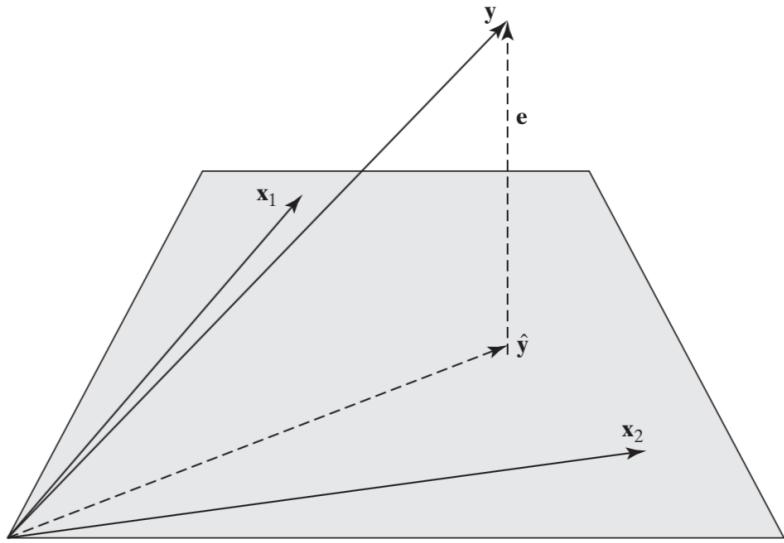


Figura 1.6: Ilustración de la poryección de y en x_1 y x_2 y el significado del error e , retomado de Greene (2012, pp. 20) [Gre12]

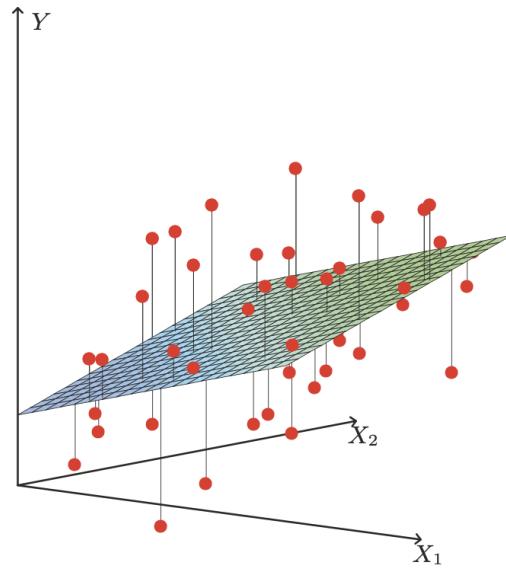


Figura 1.7: Ilustración del hiperplano generado dee regresar X_1 y X_2 en Y , retomado de Hastie, Tibshirani y Friedman (2009, pp. 45) [HTF17]

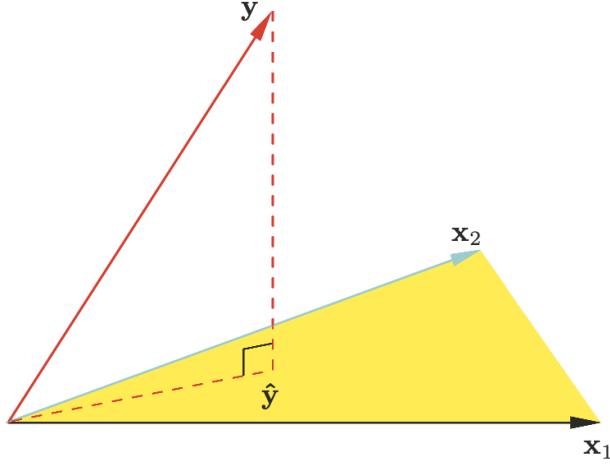


Figura 1.8: Ilustración de la poryección de y en x_1 y x_2 y el significado del error e , retomado de Hastie, Tibshirani y Friedman (2009, pp. 46) [HTF17]

Intuitivamente, la ecuación (1.7) significa que siempre que poseamos una muestra de los elementos de la población, podremos explicar una parte de la variable dependiente, no su totalidad. En este sentido, el análisis de regresión consiste en un proceso de ajuste a la variable dependiente. Esta es la idea que da origen al R^2 y otras medidas de bondad de ajuste, mismas que se analizan en textos convencionales de análisis de regresión.

El método de MCO, en consecuencia, resulta en encontrar la combinación de valores de los estimadores de los parámetros $\hat{\beta}$ que permita minimizar la suma de los residuales (estimadores de los términos de error ε) al cuadrado dada por:

$$\sum_{i=1}^N e_i^2 = \sum_{i=1}^N (y_i - \mathbf{X}'_i \hat{\beta})^2 \quad (1.8)$$

Donde $\hat{\beta}$ denota el vector de estimadores $\hat{\beta}_1, \dots, \hat{\beta}_K$. En términos matriciales, dado que $(e_1, e_2, \dots, e_n)'(e_1, e_2, \dots, e_n) = \mathbf{e}'\mathbf{e}$, el problema del método de MCO consiste en resolver el problema de optimización:

$$\begin{aligned} \text{Minimizar}_{\hat{\beta}} S(\hat{\beta}) &= \text{Minimizar}_{\hat{\beta}} \mathbf{e}'\mathbf{e} \\ &= \text{Minimizar}_{\hat{\beta}} (\mathbf{Y} - \mathbf{X}\hat{\beta})'(\mathbf{Y} - \mathbf{X}\hat{\beta}) \end{aligned}$$

Expandiendo la expresión $\mathbf{e}'\mathbf{e}$ obtenemos:

$$\mathbf{e}'\mathbf{e} = \mathbf{Y}'\mathbf{Y} - 2\mathbf{Y}'\mathbf{X}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} \quad (1.9)$$

De esta forma obtenemos que las condiciones necesarias de un mínimo son:

$$\frac{\partial S(\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}}} = -2\mathbf{X}'\mathbf{Y} + 2\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{0} \quad (1.10)$$

De ecuación anterior obtenemos para la solución del problema del mínimo a las ecuaciones siguientes conocidas como *ecuaciones normales* dadas por:

$$\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{Y} \quad (1.11)$$

Notemos que dichas ecuaciones normales son en realidad un sistema de ecuaciones de K variables o incógnitas. Por un lado, recordemos que \mathbf{X} es una matriz de dimensión $N \times K$, con lo cual \mathbf{X}' es de dimensión $K \times N$. Así, el producto $\mathbf{X}'\mathbf{X}$ dará como resultado una matriz cuadrada de dimensión $K \times K$. Por otro lado, sabemos que \mathbf{Y} es un vector de tamaño $N \times 1$, con lo cual el producto $\mathbf{X}'\mathbf{Y}$ da como resultado un vector de dimensión $K \times 1$. En conclusión, el sistema de ecuaciones normales consiste en K ecuaciones con K incógnitas $(\hat{\beta}_1, \dots, \hat{\beta}_K)$. Ante este hecho, existen múltiples formas mediante las cuales se puede solucionar dicho sistema, sin embargo en nuestro caso seguiremos el siguiente procedimiento de operaciones matriciales.

Si la matriz \mathbf{X} es de rango completo por columnas, entonces la inversa de la matriz $\mathbf{X}'\mathbf{X}$ existe. De esta forma, la solución está dada por la siguiente expresión:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \quad (1.12)$$

Finalmente, para que esta solución dada para el procedimiento de MCO y mostrada en la ecuación (1.12) sea un mínimo, debemos buscar las condiciones de segundo orden:

$$\frac{\partial^2 S(\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}} \partial \hat{\boldsymbol{\beta}}'} = 2\mathbf{X}'\mathbf{X} \quad (1.13)$$

Donde la matriz $\mathbf{X}'\mathbf{X}$ debe ser positiva definida para que la solución de MCO sea un mínimo. Sea $q = \mathbf{c}'\mathbf{X}'\mathbf{X}\mathbf{c}$ para algún vector \mathbf{c} distinto de cero. Entonces:

$$q = \mathbf{v}'\mathbf{v} = \sum_{i=1}^n v_i^2, \text{ donde } \mathbf{v} = \mathbf{X}\mathbf{c}$$

Así, q es positivo. Si \mathbf{v} fuera cero, entonces existe una combinación lineal de las columnas de \mathbf{X} que da como resultado cero, lo cual contradice el supuesto de que \mathbf{X} es de rango completo. En todos los casos, si \mathbf{X} es de rango completo, entonces la solución del método de MCO, $\hat{\boldsymbol{\beta}}$, es la única que minimiza la suma de los residuales al cuadrado.

Bondad de ajuste

Una vez planteada la solución de MCO plantearemos una medida para determinar en qué grado los datos estimados, definidos como $\hat{\mathbf{Y}} = \mathbf{X}\boldsymbol{\beta}$, se ajustan al valor real de \mathbf{Y} . La medida o métrica es el R^2 , la cual contrasta el análisis de regresión respecto de hacer una simple estimación de una media para dar un pronóstico de \mathbf{Y} . El coeficiente R^2 está montado en el supuesto de que el modelo incluye un término constante. Si el modelo no incluye una constante, no es posible hacer una interpretación del R^2 , en los siguientes párrafos abundaremos al respecto.

Antes de iniciar el desarrollo del R^2 , partamos de que la suma de residuales es igual a cero ($\sum_{i=1}^n e_i = 0$), si y sólo si, el modelo tiene un término constante. Un corolario de este hecho es que el valor promedio de los residuales es cero, sí y solo si, el modelo tiene un término constante, esto es, la única forma en que se cumpla la siguiente expresión es que la suma de residuales sea cero:

$$\begin{aligned}\bar{e} &= \frac{\sum_{i=1}^n e_i}{n} \\ &= \frac{1}{n} \cdot [1, 1, 1, \dots, 1] \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_n \end{bmatrix} \\ &= 0\end{aligned}$$

Condición que resulta de las ecuaciones normales de MCO, recordemos que ellas se derivan de que $\mathbf{X}'\mathbf{e} = \mathbf{0}$, y que el primer producto punto de vectores implica a la columna de la constante en la matriz \mathbf{X} y que está definido por $[1, 1, 1, \dots, 1]$.

El R^2 descompone la variación total en dos tramos: la variación originada por la regresión y la variación originada por el término de error. Como primer

paso definamos la variación total observada respecto de la media (\bar{Y}) como:

$$\begin{aligned}
\sum_{i=1}^n (y_i - \bar{Y})^2 &= [y_1 - \bar{Y}, \ y_2 - \bar{Y}, \ y_3 - \bar{Y}, \ \dots \ y_n - \bar{Y}] \begin{bmatrix} y_1 - \bar{Y} \\ y_2 - \bar{Y} \\ y_3 - \bar{Y} \\ \vdots \\ y_n - \bar{Y} \end{bmatrix} \\
&= (\mathbf{Y} - \bar{\mathbf{Y}})'(\mathbf{Y} - \bar{\mathbf{Y}}) \\
&= \left[\mathbf{Y} - \frac{1}{n} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \dots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} \mathbf{Y} \right]' \left[\mathbf{Y} - \frac{1}{n} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \dots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} \mathbf{Y} \right] \\
&= \mathbf{Y}' \left[\mathbb{I}_n - \frac{1}{n} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \dots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} \right]' \left[\mathbb{I}_n - \frac{1}{n} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \dots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} \right] \mathbf{Y} \\
&= \mathbf{Y}' \mathbf{M}^0 \mathbf{Y} \tag{1.14}
\end{aligned}$$

En la expresión (1.14) a \mathbf{M}^0 se puede interpretar como una matriz que resta a una matriz o vector su promedio, notemos que como resultado dará las desviaciones respecto de la media. Asimismo, \mathbf{M}^0 tiene un par de propiedades que son fácilmente demostrables: idempotencia y simetría, es decir, i) $\mathbf{M}^0 = \mathbf{M}^{0'}$ y ii) $\mathbf{M}^0 \mathbf{M}^0 = \mathbf{M}^0$. Dicho lo anterior, retomemos a (1.14) para mostrar que:

$$\begin{aligned}
\mathbf{Y}' \mathbf{M}^0 \mathbf{Y} &= \mathbf{Y}' \mathbf{M}^{0'} \mathbf{M}^0 \mathbf{Y} \\
&= (\mathbf{X} \hat{\boldsymbol{\beta}} + \mathbf{e})' \mathbf{M}^{0'} \mathbf{M}^0 (\mathbf{X} \hat{\boldsymbol{\beta}} + \mathbf{e}) \\
&= (\hat{\boldsymbol{\beta}}' \mathbf{X}' + \mathbf{e}') \mathbf{M}^{0'} \mathbf{M}^0 (\mathbf{X} \hat{\boldsymbol{\beta}} + \mathbf{e}) \\
&= \hat{\boldsymbol{\beta}}' \mathbf{X}' \mathbf{M}^{0'} \mathbf{M}^0 \mathbf{X} \hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}}' \mathbf{X}' \mathbf{M}^{0'} \mathbf{M}^0 \mathbf{e} + \mathbf{e}' \mathbf{M}^{0'} \mathbf{M}^0 \mathbf{X} \hat{\boldsymbol{\beta}} + \mathbf{e}' \mathbf{M}^{0'} \mathbf{M}^0 \mathbf{e} \\
&= \hat{\boldsymbol{\beta}}' \mathbf{X}' \mathbf{M}^{0'} \mathbf{M}^0 \mathbf{X} \hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}}' \mathbf{X}' \mathbf{e} + \mathbf{e}' \mathbf{X} \hat{\boldsymbol{\beta}} + \mathbf{e}' \mathbf{e} \\
&= \hat{\boldsymbol{\beta}}' \mathbf{X}' \mathbf{M}^{0'} \mathbf{M}^0 \mathbf{X} \hat{\boldsymbol{\beta}} + \mathbf{e}' \mathbf{e} \\
&= \hat{\boldsymbol{\beta}}' \mathbf{X}' \mathbf{M}^0 \mathbf{X} \hat{\boldsymbol{\beta}} + \mathbf{e}' \mathbf{e} \tag{1.15}
\end{aligned}$$

No debe pasar desapercibido que el desarrollo algebraico para llegar a la ecuación (1.15) sólo es posible si el promedio de los residuales es cero, es decir:

$$\mathbf{M}^0 \mathbf{e} = \mathbf{e} - \begin{bmatrix} \bar{e} \\ \bar{e} \\ \bar{e} \\ \vdots \\ \bar{e} \end{bmatrix} = \mathbf{e} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{e} \quad (1.16)$$

De otra forma, no se puede concluir la expresión (1.15). Finalmente, (1.15) la expresaremos así:

$$\mathbf{Y}' \mathbf{M}^0 \mathbf{Y} = \hat{\boldsymbol{\beta}}' \mathbf{X}' \mathbf{M}^0 \mathbf{X} \hat{\boldsymbol{\beta}} + \mathbf{e}' \mathbf{e} \quad (1.17)$$

Así, como mencionamos anteriormente, (1.17) se puede interpretar como que la variación total respecto de la media se puede descomponer en dos variaciones, una que se origina de la regresión, $\hat{\boldsymbol{\beta}}' \mathbf{X}' \mathbf{M}^0 \mathbf{X} \hat{\boldsymbol{\beta}}$, y otra de los residuales, $\mathbf{e}' \mathbf{e}$. De esta forma planteamos que R^2 es una métrica que cuantifica cuánto de la variación total es explicada por la regresión y cuánto es explicada por los residuales (es decir, por la información no observable):

$$\begin{aligned} R^2 &= \frac{\text{Var.Regresion}}{\text{Var.Total}} \\ &= \frac{\hat{\boldsymbol{\beta}}' \mathbf{X}' \mathbf{M}^0 \mathbf{X} \hat{\boldsymbol{\beta}}}{\mathbf{Y}' \mathbf{M}^0 \mathbf{Y}} \\ &= \frac{\mathbf{Y}' \mathbf{M}^0 \mathbf{Y}}{\mathbf{Y}' \mathbf{M}^0 \mathbf{Y}} - \frac{\mathbf{e}' \mathbf{e}}{\mathbf{Y}' \mathbf{M}^0 \mathbf{Y}} \\ &= 1 - \frac{\mathbf{e}' \mathbf{e}}{\mathbf{Y}' \mathbf{M}^0 \mathbf{Y}} \end{aligned} \quad (1.18)$$

Existe una expresión más que se le denomina como un R^2 ajustado o $R^2_{Adj.}$, el cual castiga por un uso excesivo de variables independientes:

$$R^2_{Adj.} = 1 - \frac{n-1}{n-K} \frac{\mathbf{e}' \mathbf{e}}{\mathbf{Y}' \mathbf{M}^0 \mathbf{Y}} \quad (1.19)$$

Así, es obvio que la siguiente relación siempre es cierta: $R^2 \geq R^2_{Adj.}$.

Inferencia asintótica bajo MCO

En estadística es común que porpongamos como estimador de σ^2 a:

$$\hat{\sigma}^2 = \frac{\mathbf{e}'\mathbf{e}}{n - K} \quad (1.20)$$

El cual es un estimador insesgado de la varianza. Además, queda pendiente en esta sección es demostrar que el estimador de MCO ($\hat{\beta}$) alcanza la cota inferior Cramér-Rao. De esta forma podríamos afirmar que el estimador es de mínima varianza o, equivalentemente, es el más eficiente.

Dicho lo anterior podemos construir las dos pruebas: t y F. Previo a dicha construcción debemos recordar que una prueba t se construye por la relación de una función normal con media cero y varianza 1 (uno), y la raíz cuadrada de una función chi-cuadrado. Supongamos que una variable Z se distribuye de forma normal con media cero y varianza 1 (uno), y que la variable χ_m^2 se distribuye de forma chi-cuadrado con m grados de libertad, entonces:

$$t = \frac{Z}{\sqrt{\frac{\chi^2}{m}}} \sim t_m \quad (1.21)$$

Donde t_m tal y como se describe en (1.21) es una distribución t de Student con m grados de libertad. Por otro lado, cuando se tiene una variable Z con distribución normal con media cero y varianza $\sigma^2 = 1$, entonces lo siguiente es cierto:

$$Z^2 \sim \chi_1^2$$

Así, la suma de variables chi-cuadrado es una chi-cuadrado en los siguientes términos:

$$\sum_{i=1}^m Z_i^2 \sim \chi_m^2$$

Dada la distribución $\hat{\beta}$, y que la varianza de cada uno de los elementos de $\hat{\beta}$, digamos $\hat{\beta}_k$, es: $\sigma^2(\mathbf{X}'\mathbf{X})_{kk}^{-1}$. En este punto denotaremos con $\hat{\beta}_k$ al elemento k -ésimo dentro de $\hat{\beta}$. Adicionalmente, con $(\mathbf{X}'\mathbf{X})_{kk}^{-1}$ al elemento en la fila k y la columna k en la matriz $(\mathbf{X}'\mathbf{X})^{-1}$. Derivado de lo anterior, lo siguiente es cierto:

$$Z = \frac{\hat{\beta}_k - \beta_k}{\sqrt{\sigma^2(\mathbf{X}'\mathbf{X})_{kk}^{-1}}} \sim \mathcal{N}(0, 1)$$

Considerando la distribución de ε_i y de que e_i es un estimador de ε_i , podemos afirmar que:

$$Z = \frac{e_i - 0}{\sigma} = \frac{e_i}{\sigma} \sim \mathcal{N}(0, 1)$$

De esta forma:

$$\sum_{i=1}^n Z_i^2 = \sum_{i=1}^n \frac{e_i^2}{\sigma^2} \sim \chi_n^2$$

Dada la distribución t de Student en la ecuación (1.21) y los elementos antes mencionados podemos llegar a las siguientes conclusiones:

$$\begin{aligned} t &= \frac{\frac{\hat{\beta}_k - \beta_k}{\sqrt{\sigma^2(\mathbf{X}'\mathbf{X})_{kk}^{-1}}}}{\sqrt{\sum_{i=1}^n \frac{e_i^2}{(n-K)\sigma^2}/(n-K)}} \\ &= \frac{\frac{\hat{\beta}_k - \beta_k}{\sqrt{\sigma^2(\mathbf{X}'\mathbf{X})_{kk}^{-1}}}}{\sqrt{\sum_{i=1}^n \frac{e_i^2}{(n-K)\sigma^2}}} \\ &= \frac{\frac{\hat{\beta}_k - \beta_k}{\sqrt{\sigma^2(\mathbf{X}'\mathbf{X})_{kk}^{-1}}}}{\sqrt{\frac{\hat{\sigma}^2}{\sigma^2}}} \\ &= \frac{\frac{\hat{\beta}_k - \beta_k}{\sqrt{\sigma^2}\sqrt{(\mathbf{X}'\mathbf{X})_{kk}^{-1}}}}{\frac{\sqrt{\hat{\sigma}^2}}{\sqrt{\sigma^2}}} \\ &= \frac{(\hat{\beta}_k - \beta_k)\sqrt{\sigma^2}}{\sqrt{\hat{\sigma}^2}\sqrt{\sigma^2}\sqrt{(\mathbf{X}'\mathbf{X})_{kk}^{-1}}} \\ &= \frac{\hat{\beta}_k - \beta_k}{\sqrt{\hat{\sigma}^2(\mathbf{X}'\mathbf{X})_{kk}^{-1}}} \end{aligned}$$

De esta forma:

$$t = \frac{\hat{\beta}_k - \beta_k}{\sqrt{\hat{\sigma}^2(\mathbf{X}'\mathbf{X})_{kk}^{-1}}} \sim t_{n-K} \quad (1.22)$$

Construida esta expresión (1.22) podemos establecer el siguiente intervalo de confianza para cada uno de nuestros estimadores $\hat{\beta}_k$:

$$-t_{\alpha/2, n-K} \sqrt{\hat{\sigma}^2(\mathbf{X}'\mathbf{X})_{kk}^{-1}} < \hat{\beta}_k - \beta_k < t_{\alpha/2, n-K} \sqrt{\hat{\sigma}^2(\mathbf{X}'\mathbf{X})_{kk}^{-1}}$$

$$\hat{\beta}_k - t_{\alpha/2,n-K} \sqrt{\hat{\sigma}^2(\mathbf{X}'\mathbf{X})_{kk}^{-1}} < \beta_k < \hat{\beta}_k + t_{\alpha/2,n-K} \sqrt{\hat{\sigma}^2(\mathbf{X}'\mathbf{X})_{kk}^{-1}}$$

Donde $t_{\alpha/2,n-K}$ es el valor crítico de tablas t-Student. Así, podemos enunciar que el estimador de MCO de la ecuación (1.12) se sujetará a la siguiente prueba de hipótesis general:

$$\begin{aligned} H_0 : \beta_k &= r \\ H_a : \beta_k &\neq r \end{aligned}$$

Donde $k = 1, 2, \dots, K$ y r es un valor respecto del cual se desea comparar a $\hat{\beta}_k$. Dicha prueba se distribuye como una t_{n-K} y su especificación más común en el análisis de regresión es la siguiente:

$$\begin{aligned} H_0 : \beta_k &= 0 \\ H_a : \beta_k &\neq 0 \end{aligned}$$

Lo que en términos de una prueba t es la siguiente:

$$t = \frac{\hat{\beta}_k - 0}{\sqrt{\hat{\sigma}^2(\mathbf{X}'\mathbf{X})_{kk}^{-1}}} \sim t_{n-k} \quad (1.23)$$

Dicha prueba la podemos interpretar como sí es posible que el estimador $\hat{\beta}_k$ sea igual a 0, o en términos económicos, si existe evidencia estadística de que el efecto de la variable x_k en y es nulo.

Como se mostró anteriormente, sólo se hemos hablado de pruebas de hipótesis cuándo se desea comprobar si alguno de los estimadores $\hat{\beta}_k$ de MCO es estadísticamente igual a algún valor determinado. No obstante, en ciertas condiciones puede ser interesante cuestionar si en conjunto todos los estimadores cumplen una cierta restricción. A este tipo de situaciones se les suele conocer como pruebas de hipótesis globales. Sin pérdida de generalidad podemos enunciar a una prueba global como:

$$\begin{aligned} H_0 : \mathbf{R}\boldsymbol{\beta} &= \mathbf{r} \\ H_a : \mathbf{R}\boldsymbol{\beta} &\neq \mathbf{r} \end{aligned}$$

Donde, β es el vector de parámetros del MCO de la ecuación (1.12), \mathbf{R} y \mathbf{r} son de la siguiente manera:

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} & \dots & R_{1K} \\ R_{21} & R_{22} & \dots & R_{2K} \\ R_{31} & R_{32} & \dots & R_{3K} \\ \vdots & \vdots & \dots & \vdots \\ R_{J1} & R_{J2} & \dots & R_{JK} \end{bmatrix}$$

En \mathbf{R} cada una de las R_{jk} toma el valor de 1 o 0, según sea la estructura de la restricción. Con $j = 1, 2, \dots, J$. En el caso de \mathbf{r} :

$$\mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_J \end{bmatrix}$$

Donde cada uno de los r_j representa el valor de la restricción que es evaluada. No debe pasar desapercibido que en este caso existen J restricciones ($J \leq K$), mismas que se evalúan de forma conjunta en una sola prueba. Esa es la ventaja de la prueba que se enuncia a continuación.

Definamos a F de Fisher como la razón de dos pruebas t de Student, la primera para los valores de una combinación lineal de β , y la segunda para el estimador de la varianza $\hat{\sigma}^2$:

$$\begin{aligned} F &= \frac{(\mathbf{R}\hat{\beta} - \mathbf{r})'[\mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}']^{-1}(\mathbf{R}\hat{\beta} - \mathbf{r})/J}{\hat{\sigma}^2} \\ &= \frac{(\mathbf{R}\hat{\beta} - \mathbf{r})'[\mathbf{R}\hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}']^{-1}(\mathbf{R}\hat{\beta} - \mathbf{r})}{J} \\ &= \frac{(\mathbf{R}\hat{\beta} - \mathbf{r})'[\mathbf{R}Var(\hat{\beta})\mathbf{R}']^{-1}(\mathbf{R}\hat{\beta} - \mathbf{r})}{J} \end{aligned}$$

Por lo tanto la estadística F de prueba será:

$$F = \frac{(\mathbf{R}\hat{\beta} - \mathbf{r})'[\mathbf{R}\hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}']^{-1}(\mathbf{R}\hat{\beta} - \mathbf{r})}{J} \sim F_{J,n-K} \quad (1.24)$$

Así, la prueba de hipótesis que es la más común en el análisis de regresión y que se le conoce como prueba global. Dicha prueba asume una forma de la

matriz \mathbf{R} definida así:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

Y un vector \mathbf{r} :

$$\mathbf{r} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

De esta forma la hipótesis nula de una prueba global se puede escribir como:

$$\begin{aligned} H_0 : \quad \beta_1 &= \beta_2 = \dots = \beta_K = 0 \\ H_a : \quad \text{No } H_0 \end{aligned}$$

Esta prueba se le conoce como prueba global, ya que prueba si en conjunto todas las variables independientes tienen un efecto nulo en \mathbf{Y} .

En este caso nos referimos a casos como:

$$H_0 : \mathbf{R}\boldsymbol{\beta} = r$$

Donde \mathbf{R} es una matriz de restricciones de dimensión $Q \times K$, con $Q \leq K$, r es un vector de dimensión $Q \times 1$. Así en casos de heterocedasticidad utilizaremos una prueba de Wald dada por:

$$\mathbf{W} = (\mathbf{R}\hat{\boldsymbol{\beta}} - r)'(\mathbf{R}\hat{\mathbf{V}}\mathbf{R}')^{-1}(\mathbf{R}\hat{\boldsymbol{\beta}} - r)$$

Donde $\hat{\mathbf{V}} = \widehat{AVar}(\hat{\boldsymbol{\beta}})$, por lo que la distribución de \mathbf{W} será como:

$$\mathbf{W} \sim \chi_Q^2$$

Lo que implica que la formulación $F_{Q,N-K}$ no es de aplicación.

Otro tipo de pruebas que pueden realizarse son aquellas basadas en Multiplicadores de Lagrange. Supongamos un modelo lineal particionado de la siguiente manera:

$$y_i = \mathbf{x}_{i1}\boldsymbol{\beta}_1 + \mathbf{x}_{i2}\boldsymbol{\beta}_2 + \varepsilon \quad (1.25)$$

Supongamos que desemos probar la hipótesis de que:

$$H_0 : \beta_2 = 0$$

Sea $\tilde{\beta}_1$ esl estimador de β_1 bajo la hipótesis nula anterior, digamos que este estimador corresponde al modelo restringido. Definamos los residuales:

$$e_i = y_i - \mathbf{x}_{i1}\tilde{\beta}_1$$

Bajo la hiótesis nula, \mathbf{x}_{i2} debería ser una varible no correlacionada con e_i . En este caso la prueba estadística es obtenida como sigue:

1. Corremos una regresión de e_i en \mathbf{x}_{i1} y \mathbf{x}_{i2}
2. Asumiendo que \mathbf{x}_{i1} contiene término constante, digamos que R_E^2 denota al R^2 asociado a la regresión.
3. Entonces propongamos que la estadística de Multiplicadores de Lagrange es:

$$LM = N \cdot R_E^2$$

Para llgar a la expresión anterior se requiere de solucionar un problema de optimización.

4. Bajo la H_0 , tenemos que:

$$LM \sim \chi_{K_2}^2$$

Donde K_2 es el número de restricciones que hemos probado. De esta forma, si $N \cdot R_E^2$ es suficientemente grande, entonces e_i estaba significativamente correlacionado con \mathbf{x}_{i2} , por lo que la hipótesis debió ser rechazada.

En este punto debe quedar claro que R^2 requiere que $\sum_{i=1}^N e_i = 0$, lo que implica que el modelo tiene un término constante.

1.2.2. Modelos no lineales

¿Qué pasa cuando la relación observada entre las variables no es lineal? Casi siempre pasa eso. A continuación discutiremos tres casos de ecuaciones que nos ayudaran a describir una relación no lineal mediante una transformación que hace la relaciones lineales. Cada uno responde a condiciones o características que la experiencia del investigador tiene que solventar.

Cada una de las ecuaciones se distingue por la tasa de cambio de la variable y en relación a los cambios de la variable x . Partamos de que la forma más simple de una relación lineal está dada por la ecuación (1.26). Note que hemos eliminado los subíndices para indicar que la ecuación se cumple para cada uno de los elementos en la muestra, $i = 1, 2, \dots, n$. Sin pérdida de generalidad digamos que en lugar de β_1 y β_2 empleamos a y b como constantes, como una forma de ilustrar cada uno de los fenómenos que discutimos a continuación.

$$y = a + bx \quad (1.26)$$

Esta ecuación (1.26) tiene la característica de que tiene una tasa de variación constante para y en relación a x , de esta forma:

$$\frac{dy}{dx} = b \iff dy = bdx \iff \int dy = \int bdx \iff y = a + bx$$

Los siguientes modelos no lineales tienen una motivación similar, dependen de la forma en que se asuma la variación de y en función de la variación de x .

Regresión exponencial

Supongamos que y depende de x y que los cambios en y derivados de x son proporcionales a y , es decir:

$$\frac{dy}{dx} = by$$

Donde b es una constante. De esta forma podemos encontrar que:

$$\frac{dy}{dx} = by \iff \frac{dy}{y} = bdx \iff \int \frac{dy}{y} = \int bdx \iff \ln(y) = a + bx$$

De esta forma, tenemos que:

$$y = e^{bx} e^a \iff y = k e^{bx}$$

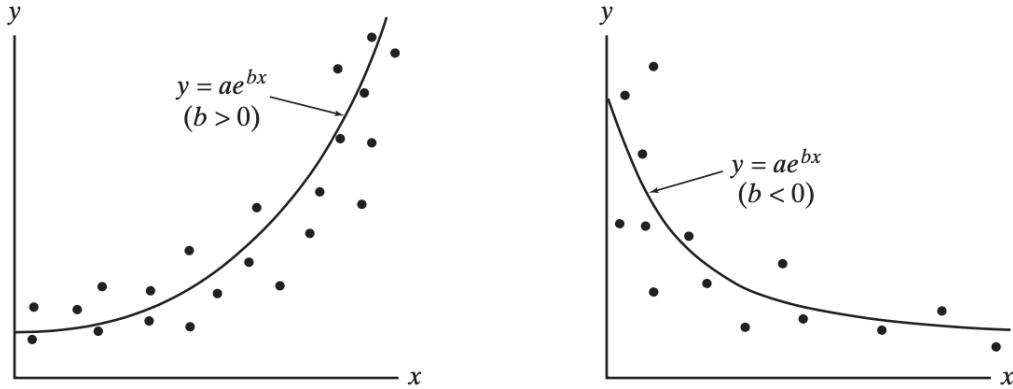


Figura 1.9: Funciones exponenciales, Retomado de Larsen y Marx (2012, p. 532) [LM12]

Por lo tanto, cuando estimamos una relación: $\ln(y) = a + bx$ estamos asumiendo que existe la siguiente relación: $\frac{dy}{dx} = by$. A este tipo de ecuaciones se les conoce como log-lineales. En la Figura 1.9 se ilustra la relación log-lineal - exponencial.

Regresión logarítmica

En este caso suponemos que los cambios en y por causa de cambios en x son proporcionales a la razón que guardan y y x , es decir:

$$\frac{dy}{dx} = b \frac{y}{x}$$

Por lo que podemos establecer:

$$\frac{dy}{dx} = b \frac{y}{x} \iff \frac{dy}{y} = b \frac{dx}{x} \iff \int \frac{dy}{y} = \int b \frac{dx}{x} \iff \ln(y) = a + \ln(x)$$

Así, podemos establecer que la forma funcional de la relación de y y x es como sigue y como se muestra en la Figura 1.10.

$$y = kx^b$$

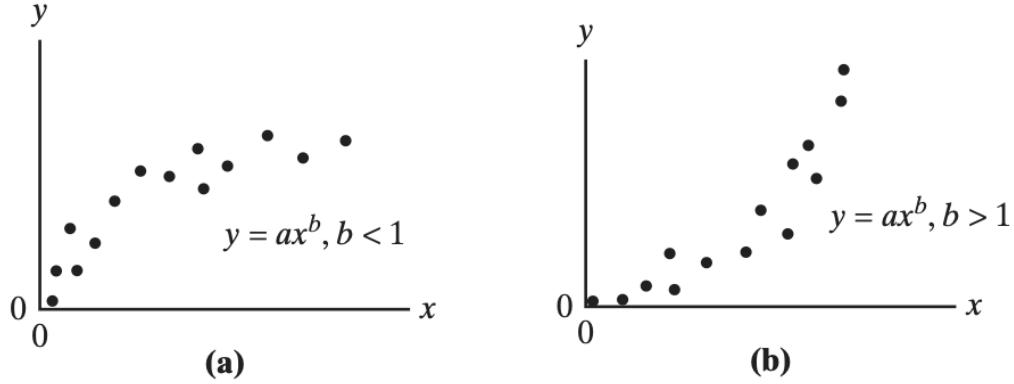


Figura 1.10: Funciones logarítmicas, Retomado de Larsen y Marx (2012, p. 536) [LM12]

Funciones logísticas

Este tipo de ecuaciones tienen la característica de que permiten modelar crecimientos poblacionales, aceptación de políticas públicas, aceptación de tecnologías, epidemias, etc. En este caso suponemos que enfrentamos que los cambios en y dados los cambios en x son proporcionales a y y a las distancia que y tiene respecto de un factor de saturación L (límite superior o punto de saturación poblacional, cobertura universal, infecciones en el total de la población, etc.).

En este caso suponemos que la variación estará dada por:

$$\frac{dy}{dx} = ky(L - y)$$

Donde k y L son constantes. Una vez solucionada la ecuación diferencial anterior, encontraremos que, si $L = 1$, entonces la solución será como sigue, la Figura 1.11 ilustra ecuaciones como la siguiente:

$$y = \frac{1}{1 + e^{a+bx}} \quad (1.27)$$

Ahora mostremos como llegar a la solución de la ecuación logística:

$$\frac{dy}{dx} = ky(L - y) \iff \frac{dy}{y(L - y)} = kdx \iff \left(\frac{\frac{1}{L}}{y} + \frac{\frac{1}{L}}{(L - y)} \right) dy = kdx$$

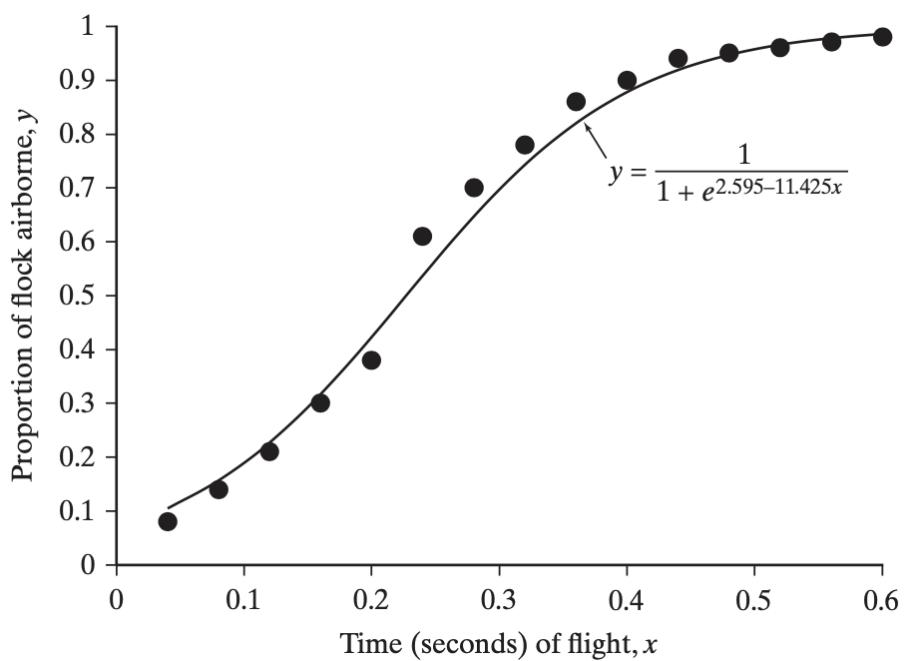


Figura 1.11: Funciones logísticas, Retomado de Larsen y Marx (2012, p. 540)
[LM12]

$$\begin{aligned}
&\iff \frac{1}{L} \left(\frac{dy}{y} + \frac{dy}{(L-y)} \right) = kdx \iff \int \frac{1}{L} \left(\frac{dy}{y} + \frac{dy}{(L-y)} \right) = \int kdx \\
&\iff \left(\int \frac{dy}{y} + \int \frac{dy}{(L-y)} \right) = \int Lkdx \iff (\ln(y) - \ln(L-y)) = Lkx + C \\
&\iff (\ln(L-y) - \ln(y)) = -LC - Lkx \iff \ln \left(\frac{L-y}{y} \right) = -LC - Lkx \\
&\iff \left(\frac{L-y}{y} \right) = e^{-LC-Lkx} \iff y = \frac{L}{1 + e^{-LC-Lkx}}
\end{aligned}$$

Por lo tanto, la solución general de la ecuación será:

$$y = \frac{L}{1 + e^{a+bx}}$$

Donde asumimos que la forma lineal de la ecuación logística es:

$$\ln \left(\frac{L-y}{y} \right) = a + bx$$

Funciones polinomiales

Otro tipo de ecuaciones o funciones se asocian con una relación entre y y x del tipo no lineal, pero que puede describirse bajo una formulación de un polinomio de algún grado l . Este tipo de ecuaciones tienen la característica de que permiten modelar crecimientos poblacionales, aceptación de políticas públicas, aceptación de tecnologías, epidemias, etc., mediante una especificación menos compleja en términos matemáticos.

En muchas aplicaciones, una simple transformación logarítmica de la variable x ya no puede ser suficiente para lograr un ajuste satisfactorio. Una forma sencilla de obtener más flexibilidad es utilizar la regresión polinomial. Normalmente, se prefieren polinomios de bajo grado l . En la práctica, rara vez utilizamos polinomios de grado superior a $l = 3$, ya que las estimaciones pueden volverse bastante inestables con una alta variabilidad en los límites

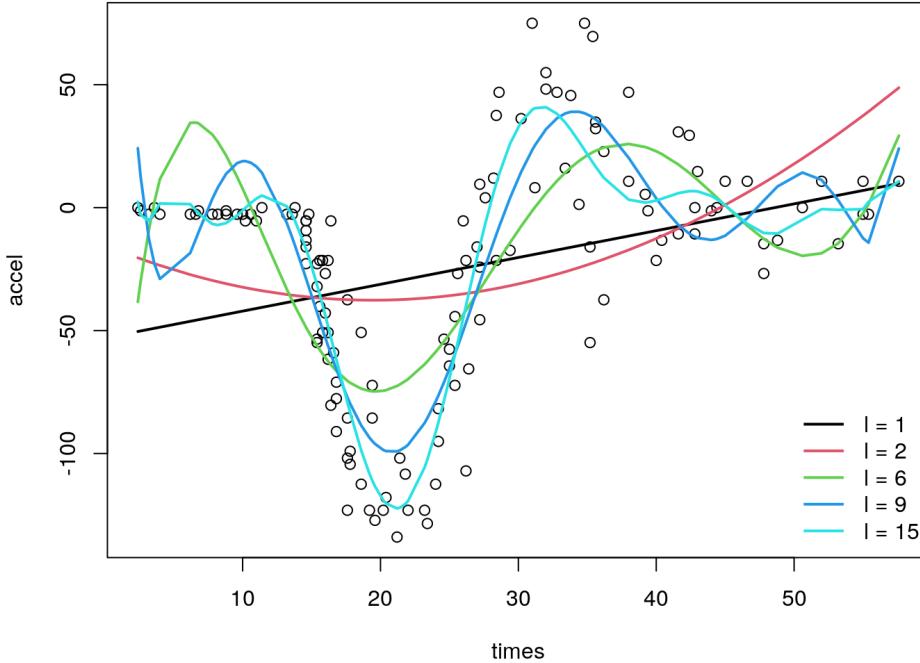


Figura 1.12: Funciones polinomiales, retomado de <https://discdown.org/flexregression/smoothreg.html>

del dominio de variables. Además, pueden surgir inestabilidades numéricas en grados muy altos, en cuyo caso se requiere de re-escalar los datos.

En general partamos de ecuaciones como:

$$y = a + b_1x^1 + b_2x^2 + \dots + b_lx^l \quad (1.28)$$

La figura 1.12 ilustra diferentes escenarios de ecuaciones como 1.28 con diferentes grados l .

1.2.3. Introducción al Aprendizaje Estadístico

El aprendizaje estadístico juega un rol esencial en muchas áreas de la ciencia, finanzas y la industria. Algunos ejemplos son:

1. Predecir si un paciente—que se encuentra hospitalizado debido a un ataque al corazón—tenrá un segundo ataque. La predicción estará basada en métricas demográficas, de la dieta y de registros clínicos.

2. Predecir el precio de una acción en los siguientes 6 meses; considerando la base de las medidas de desempeño de la compañía y de otros datos económicos.
3. Identificar los números en la digitalización de formas escritas a mano.
4. Identificar los factores de riesgo para el cáncer de próstata, basados en datos clínicos y de otras variables demográficas.

El aprendizaje estadístico comprende a un conjunto de herramientas para modelar y entender conjuntos de datos complejos. También se le conoce como Machine Learning (ML), el cual conjuga el desarrollo reciente en el área de la estadística junto con el crecimiento en paralelo de la computación.

El aprendizaje estadístico considera muchos métodos convencionales y de uso amplio como análisis de regresión, clasificación, árboles de decisión, etc. También se refiere a una amplia gama de herramientas para entender o interpretar datos clasificadas como supervisadas y no supervisadas.

- El aprendizaje estadístico supervisado involucra la construcción de un modelo estadístico para predecir o estimar un resultado (o variable que se pide supervisar) basado en uno o más variables explicativas.
- El aprendizaje no supervisado considera variables explicativas, pero los resultados observados no son una variable explicada que sea susceptible de supervisión.

Ahora introduzcamos un poco de notación. Denotaremos a una variable independiente o explicativa con x_{ij} , si dicha variable es un conjunto de variables acomodadas en un vector utilizaremos \mathbf{X}_i ; en estos casos denotaremos a un elemento o variable del vector \mathbf{X}_i como x_{ik} , donde $i = 1, 2, \dots, n$ denota a los individuos en la muestra y $k = 1, 2, \dots, K$ denota al número de variables. Por convención diremos que $x_{i1} = 1$ para todo $i = 1, 2, \dots, n$, ya que en dicha variable consideraremos al término constante en la regresión.

Por su parte, los resultados, variables dependientes o variables de respuesta se denominarán como:

- y_i denotará una respuesta que es una cantidad continua
- g_i denotará una respuesta cualitativa, discreta o de grupo

Así, con \mathbf{X} , y \mathbf{Y} y \mathbf{G} denotaremos al a la matriz y vector columna que contiene a todos los valores de las variables dependientes y de respuesta apiladas para cada uno de los elementos en la muestra indexados con i .

Usaremos mayúsculas como X , Y o G para representar a los aspectos genéricos de las variables. De esta forma, debe ser claro que las letras minúsculas serán empleadas para representar a valores observados de las variables, así el valor observado de la variable k -ésima para el elemento de la muestra i -ésimo en \mathbf{X} será representado como x_{ij} . Finalmente, con la notación \hat{Y} o \hat{G} representaremos a los valores estimados o predecidos.

El aprendizaje estadístico parte del establecimiento de que una variable dependiente Y es una función de un conjunto de variables explicativas $\mathbf{X} = [X_1, X_2, \dots, X_p]$. De esta forma platearemos:

$$Y = f(\mathbf{X}) + \varepsilon \quad (1.29)$$

Así, el aprendizaje estadístico se trata de un conjunto de aproximaciones para f . ¿Para qué estimar f ? La respuesta es para hacer predicciones y para hacer inferencia. La más común de ambas razones es la predicción, con una predicción de Y podríamos establecer:

$$\hat{Y} = \hat{f}(\mathbf{X}) \quad (1.30)$$

Ajuste y separación del conjunto de datos

Supongamos la variable objetivo Y , un vector de variables explicativas o variables 'input' X y un modelo predictivo $\hat{f}(X)$ que es estimado a partir de un conjunto de entrenamiento τ .

Definiremos la función de pérdida derivada de la estimación y capturada por los errores entre Y y $\hat{f}(X)$ estará dada por:

$$L(Y, \hat{f}(X)) = \begin{cases} (Y - \hat{f}(X))^2 & \text{error cuadrático} \\ |Y - \hat{f}(X)| & \text{error absoluto} \end{cases} \quad (1.31)$$

De esta forma podemos establecer un error cuadrático como:

$$\begin{aligned} \mathbb{E}[Y - \hat{Y}]^2 &= \mathbb{E}[f(X) + \varepsilon - \hat{f}(X)]^2 \\ &= \mathbb{E}[f(X) - \hat{f}(X)]^2 + Var(\varepsilon) \end{aligned}$$

Donde $\mathbb{E}[f(X) - \hat{f}(X)]^2$ es el único componente reducible. Para hacer predicciones requerimos de un connjunto de datos de entrenamiento y otro



Figura 1.13: División del conjunto de datos, retomado de Hastie, Tibshirani, y Friedman (2017, p. 222) [HTF17]

más de prueba – en el primero estimamos $f(\cdot)$ y en el segundo hacemos predicciones –.

Finalmente, la condición de inferencia nos permite construir pruebas de hipótesis, estimadores que cumple ciertas propiedades, intervalos de confianza, etc. Así, dividiremos al conjunto de datos conforme se describe en la Figura 1.13.

Para proporcionar una comprensión precisa de la generalización de nuestro modelo óptimo final, se suele dividir nuestros datos sólo en dos conjuntos de datos: de entrenamiento y de prueba

- Conjunto de entrenamiento: estos datos se utilizan para desarrollar conjuntos de características, entrenar nuestros algoritmos, ajustar hiperparámetros, comparar modelos y todas las demás actividades necesarias para elegir un modelo final (por ejemplo, el modelo que queremos poner en producción).
- Conjunto de pruebas: una vez elegido un modelo final, estos datos se utilizan para estimar una evaluación insesgada del rendimiento del modelo, a lo que nos referimos como error de generalización.

1.2.4. 4.1.1 Regresión lineal múltiple con datos de publicidad en medios de comunicación.

Ver RMarkdown: 01_Ejemplo_Datos_Publicidad.Rmd

1.2.5. Método de regresiones restringidas (Shrinkage methods).

Este tipo de métodos permiten (1) restringir los posibles valores de las estimaciones del parámetro β a un subconjunto y (2) seleccionar variables o regresores dentro de un conjunto de candidatos. Veámos dos casos particulares

i) una regresión de cresta destringida (ridge regression) y ii) la regresión Lasso (least absolute shrinkage and selection operator).

Regresión Ridge.

La regresión de ridge restringe los coeficientes de la regresión mediante la imposición de una penalización en su magnitud. Los coeficientes estimados por este método resultan de resolver el problema de minimizar los residuales al cuadrado sujeto a que los coeficientes sumen un valor dado. Es decir:

$$\hat{\boldsymbol{\beta}}^{Ridge} = \min_{\boldsymbol{\beta}} \left[\sum_{i=1}^N (y_i - \mathbf{X}'_i \boldsymbol{\beta})^2 + \lambda \sum_{k=2}^K \beta_k^2 \right] \quad (1.32)$$

Donde $\lambda \geq 0$ y suponemos que β_1 es el término constante de la regresión, el cual no se optimiza.

La idea de la penalización de los parámetros se deriva de que la forma equivalente del problema en la ecuación (1.32) es:

$$\hat{\boldsymbol{\beta}}^{Ridge} = \min_{\boldsymbol{\beta}} \left[\sum_{i=1}^n (y_i - \mathbf{X}_i \boldsymbol{\beta})^2 \right]$$

Sujeto a:

$$\sum_{k=2}^K \beta_k^2 \leq t$$

Note que la penalización no aplica al término constante, ¿por qué?, para garantizar que la estimación del hiperplano asociado pasa por la media de Y y no por el cero (0).

¿Qué implicaciones tiene para la estimación? Es posible estimar el modelo de la ecuación (1.32) pero ajustando las variables incluidas con la resta de la media. Esto, ya que el término constante se puede estimar considerando que este toma el valor de la media de Y dado por:

$$\bar{Y} = \frac{\sum_{i=1}^n y_i}{n}$$

Para los restantes coeficientes los determinaremos mediante un procedimiento dado por:

$$\min_{\boldsymbol{\beta}_R, \lambda} [(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}_R)'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}_R) + \lambda \boldsymbol{\beta}'_R \boldsymbol{\beta}_R] \quad (1.33)$$

Donde β_R contiene sólo las pendientes. Para determinar un valor estimado debemos resolver el problema descrito en la ecuación (1.33):

$$\begin{aligned}\frac{\partial}{\partial \beta_R} S(\beta_R) &= \frac{\partial}{\partial \beta_R} ((\mathbf{Y} - \mathbf{X}\beta_R)'(\mathbf{Y} - \mathbf{X}\beta_R) + \lambda\beta_R'\beta_R) \\ &= \frac{\partial}{\partial \beta_R} (\mathbf{Y}'\mathbf{Y} - 2\mathbf{Y}'\mathbf{X}\beta_R + \beta_R'\mathbf{X}'\mathbf{X}\beta_R + \lambda\beta_R'\beta_R) \\ &= -2\mathbf{X}'\mathbf{Y} + 2\mathbf{X}'\mathbf{X}\beta_R + 2\lambda\beta_R\end{aligned}$$

Determinando el mínimo:

$$\begin{aligned}-2\mathbf{X}'\mathbf{Y} + 2\mathbf{X}'\mathbf{X}\hat{\beta}_R^{Ridge} + 2\lambda\hat{\beta}_R^{Ridge} &= 0 \\ -\mathbf{X}'\mathbf{Y} + (\mathbf{X}'\mathbf{X} + \lambda\mathbb{I}_{K-1})\hat{\beta}_R^{Ridge} &= 0\end{aligned}$$

Por lo tanto, el estimador estará dado por:

$$\hat{\beta}_R^{Ridge} = (\mathbf{X}'\mathbf{X} + \lambda\mathbb{I}_{K-1})^{-1}\mathbf{X}'\mathbf{Y} \quad (1.34)$$

Así, el problema de la estimación es que tiene 1 grado de libertad, λ , que es un valor de penalización y que resulta arbitrario-determinado de forma iterativa hasta alcanzar uno que minimice la ecuación (1.32).

Regresión Lasso (Least Absolute Shrinkage and Selection Operator).

La regresión Lasso se define por la solución al problema:

$$\hat{\beta}^{Lasso} = \min_{\beta} \left[\sum_{i=1}^n (y_i - \mathbf{X}_i\beta)^2 \right] \quad (1.35)$$

Sujeto a:

$$\sum_{k=2}^K |\beta_k| \leq t$$

Donde $\lambda \geq 0$ y suponemos que β_1 es el término constante de la regresión. El término constante se estima considerando que este toma el valor de la media de Y dado por:

$$\bar{Y} = \frac{\sum_{i=1}^n y_i}{n}$$

La idea de la penalización de los parámetros se deriva de la forma equivalente del problema en la ecuación (1.35) es:

$$\hat{\boldsymbol{\beta}}^{Lasso} = \min_{\boldsymbol{\beta}} \left[\sum_{i=1}^n (y_i - \mathbf{X}_i \boldsymbol{\beta})^2 + \lambda \sum_{k=2}^K |\beta_k| \right] \quad (1.36)$$

1.2.6. 4.1.2 Regresión lineal con métodos de restricción (Shrinkage Methods) y regularización de los parámetros.

Ver Python Notebook: 02_Ejemplo_Datos_Inmuebles24.ipynb

1.3. 4.2 Aplicación de modelos de regresión de respuesta binaria.

1.3.1. Métodos de estimación basados en verosimilitud Introducción

En esta sección discutiremos el tratamiento general para la estimación por máxima verosimilitud. La diferencia significativa en estos casos es que asumiremos una distribución subyacente. Noten que los métodos utilizados anteriormente no requieren de que asumamos una distribución particular para realizar la estimación.

El planteamiento general del método de máxima verosimilitud considera un conjunto de observaciones independientes e idénticamente distribuidas descritas por la familia: $\{y_i, \mathbf{x}_i : y_i \in \mathbb{R}, \mathbf{x}_i \in \mathbb{R}^K\}$.

En estos casos también supondremos que buscamos estimar la regresión de y_i es \mathbf{x}_i y no el caso contrario. Por lo tanto, el método requiere de la función de densidad condicional de y_i en \mathbf{x}_i , es decir:

$$f(y_i | \mathbf{x}_i)$$

Así, al método en realidad se le conoce como método de máxima verosimilitud condicional. Su aplicación es amplia en casos como:

1. Modelos lineales: regresión de una sola ecuación, sistemas de ecuaciones, modelos de datos panel y modelos generalizados con heterocedasticidad y autocorrelación

2. Modelos no lineales: respuesta binaria, respuesta categórica, respuesta multinivel
3. Modelos de conteo y duración

Ejemplo. Sea y_i^* una variable aleatoria latente que se pueda expresar como:

$$y_i^* = \mathbf{x}_i \boldsymbol{\theta} + \varepsilon_i$$

Donde ε_i es independiente de \mathbf{x}_i , el cual es un vector de $1 \times K$ que contienen un término constante. $\boldsymbol{\theta}$ es un vector de $K \times 1$ parámetros a estimar. Finalmente, asumiremos que:

$$\varepsilon_i \sim Normal(0, 1)$$

Supongamos que en lugar de y_i^* nosotros sólo podemos observar la variable binaria indicadora:

$$y_i = \begin{cases} 1 & \text{si } y_i^* > 0 \\ 0 & \text{si } y_i^* \leq 0 \end{cases}$$

En este caso, la pregunta que pretendemos responder sería: ¿cuál es la probabilidad de que $y_i = 1$ condicional en que tenemos el vector \mathbf{x}_i ?

$$\begin{aligned} P(y_i = 1) &= P(y_i^* > 0 | \mathbf{x}_i) \\ &= P(\mathbf{x}_i \boldsymbol{\theta} + \varepsilon_i > 0 | \mathbf{x}_i) \\ &= P(\varepsilon_i > -\mathbf{x}_i \boldsymbol{\theta} | \mathbf{x}_i) \end{aligned}$$

Dada la distribución ε_i , entonces:

$$P(\varepsilon_i > -\mathbf{x}_i \boldsymbol{\theta} | \mathbf{x}_i) = \int_{-\mathbf{x}_i \boldsymbol{\theta}}^{\infty} f(\varepsilon_i | \mathbf{x}_i) d\varepsilon_i$$

Sabemos que toda función de densidad cumple con:

$$\begin{aligned} 1 &= \int_{-\infty}^{\infty} f(\varepsilon_i | \mathbf{x}_i) d\varepsilon_i \\ &= \int_{-\infty}^{-\mathbf{x}_i \boldsymbol{\theta}} f(\varepsilon_i | \mathbf{x}_i) d\varepsilon_i + \int_{-\mathbf{x}_i \boldsymbol{\theta}}^{\infty} f(\varepsilon_i | \mathbf{x}_i) d\varepsilon_i \\ &= \int_{-\infty}^{-\mathbf{x}_i \boldsymbol{\theta}} f(\varepsilon_i | \mathbf{x}_i) d\varepsilon_i + \int_{-\infty}^{\mathbf{x}_i \boldsymbol{\theta}} f(\varepsilon_i | \mathbf{x}_i) d\varepsilon_i \\ &= \Phi(-\mathbf{x}_i \boldsymbol{\theta}) + \Phi(\mathbf{x}_i \boldsymbol{\theta}) \\ &= (1 - \Phi(\mathbf{x}_i \boldsymbol{\theta})) + \Phi(\mathbf{x}_i \boldsymbol{\theta}) \\ &= P(\varepsilon_i < -\mathbf{x}_i \boldsymbol{\theta} | \mathbf{x}_i) + P(\varepsilon_i > -\mathbf{x}_i \boldsymbol{\theta} | \mathbf{x}_i) \end{aligned}$$

De esta forma tenemos dos ecuaciones:

$$\begin{aligned} P(y_i = 1|\mathbf{x}_i) &= P(\varepsilon_i > -\mathbf{x}_i \boldsymbol{\theta}|\mathbf{x}_i) = \Phi(\mathbf{x}_i \boldsymbol{\theta}) \\ P(y_i = 0|\mathbf{x}_i) &= P(\varepsilon_i < -\mathbf{x}_i \boldsymbol{\theta}|\mathbf{x}_i) = 1 - \Phi(\mathbf{x}_i \boldsymbol{\theta}) \end{aligned}$$

Estas dos funciones de probabilidad son independientes. De esta forma, la función de densidad condicional de y_i será:

$$\begin{aligned} f(y_i|\mathbf{x}_i) &= P(y_i = 1|\mathbf{x}_i)^{y_i} \cdot P(y_i = 0|\mathbf{x}_i)^{1-y_i} \\ &= \Phi(\mathbf{x}_i \boldsymbol{\theta})^{y_i} \cdot (1 - \Phi(\mathbf{x}_i \boldsymbol{\theta}))^{1-y_i} \end{aligned}$$

Marco general de estimación por máxima verosimilitud

Sea $f(y_i|\mathbf{x}_i)$ la función de densidad condicional de y_i dado \mathbf{x}_i . Sea $\boldsymbol{\theta}$ un conjunto de parámetros de la función. Entonces la función de densidad conjunta de variables aleatorias independientes $\{y_i : y_i \in \mathbb{R}\}$ dados los valores $\{\mathbf{x}_i : \mathbf{x}_i \in \mathbb{R}^K\}$ estará dada por:

$$\prod_{i=1}^n f(y_i|\mathbf{x}_i; \boldsymbol{\theta}) = f(y_1, y_2, \dots, y_n | \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n; \boldsymbol{\theta}) = L(\boldsymbol{\theta}) \quad (1.37)$$

A la ecuación (1.37) se le conoce como ecuación de verosimilitud. El problema de máxima verosimilitud entonces será:

$$\max_{\boldsymbol{\theta} \in \Theta} \prod_{i=1}^n f(y_i|\mathbf{x}_i; \boldsymbol{\theta}) = \max_{\boldsymbol{\theta} \in \Theta} L(\boldsymbol{\theta}) \quad (1.38)$$

Dado que el logaritmo natural es una transformación monótona, podemos decir que el problema de la ecuación (1.38) es equivalente a:

$$\max_{\boldsymbol{\theta} \in \Theta} \ln L(\boldsymbol{\theta}) = \max_{\boldsymbol{\theta} \in \Theta} \ln \prod_{i=1}^n f(y_i|\mathbf{x}_i; \boldsymbol{\theta}) = \max_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^n \ln f(y_i|\mathbf{x}_i; \boldsymbol{\theta}) \quad (1.39)$$

Para solucionar el problema se tiene que determinar las condiciones de primer y segundo orden, las cuales serán:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \ln L(\boldsymbol{\theta}) = \nabla \ln L(\boldsymbol{\theta}) \quad (1.40)$$

$$\frac{\partial^2}{\partial^2 \boldsymbol{\theta}} \ln L(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \ln L(\boldsymbol{\theta}) \cdot \frac{\partial}{\partial \boldsymbol{\theta}} \ln L(\boldsymbol{\theta}') = H(\boldsymbol{\theta}) \quad (1.41)$$

La solución estará dada por aquel valor de $\hat{\boldsymbol{\theta}}$ que hace:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \ln L(\hat{\boldsymbol{\theta}}) = 0$$

A su vez, la varianza será aquella que resulta de:

$$Var[\hat{\boldsymbol{\theta}} | \mathbf{X}] = (-\mathbb{E}_{\hat{\boldsymbol{\theta}}}[H(\boldsymbol{\theta})])^{-1}$$

Pruebas de hipótesis

1. Para el caso de pruebas de hipótesis sobre cada uno de los coeficientes estimados $\hat{\theta}_i$, dada la construcción de la función de verosimilitud y que se trata de grandes muestras, la función de distribución de ese coeficiente sera una normal, por lo que las hipótesis planteadas serán asumiendo:

$$Z = \frac{\hat{\theta}_i - \theta_i}{\sqrt{Var(\hat{\theta}_i)}} \sim N(0, Var(\hat{\theta}_i)) \quad (1.42)$$

2. Para pruebas globales utilizaremos la prueba de Wald:

$$W = c(\hat{\boldsymbol{\theta}})' (\hat{C} Var(\hat{\boldsymbol{\theta}}) \hat{C}')^{-1} c(\hat{\boldsymbol{\theta}}) \sim \chi^2_Q \quad (1.43)$$

Donde $c(\hat{\boldsymbol{\theta}})$ es una función de los coeficientes, \hat{C} es el jacobiano de $c(\hat{\boldsymbol{\theta}})$, Q es el número de restricciones.

3. Alternativamente, para pruebas sobre los coeficientes podremos utilizar pruebas de razón de verosimilitud:

$$LR = 2[\ln L(\hat{\boldsymbol{\theta}}_0) - \ln L(\hat{\boldsymbol{\theta}}_R)] \sim \chi^2_Q \quad (1.44)$$

1.3.2. Estimación de modelos no lineales

Modelos de respuesta binaria

Planteamiento general. En el caso de modelos de respuesta binaria, asumiremos que y_i es una variable aleatoria que toma sólo valores de 0 y 1. Los ejemplos pueden ser amplios. Decisiones sobre consumir o no, pagar o no, ir a una escuela determinada, viajar por tierra o aire. En general diremos que 1 es éxito y 0 es fracaso.

Como en el caso de modelos lineales diremos que y_i es la variable dependiente y $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iK})$ un vector de variables independientes o explicativas. Así, en los modelos no lineales de respuesta binaria estaremos interesados en la probabilidades:

$$P(y_i = 1|\mathbf{x}_i) = P(y_i = 1|x_{i1}, x_{i2}, \dots, x_{iK})$$

Dicho lo anterior y de forma similar al de la regresión lineal, podemos establecer el efecto marginal para el caso de una variable independiente x_j continua:

$$EMg_j = \frac{\partial}{\partial x_j} P(y_i = 1|x_{i1}, x_{i2}, \dots, x_{iK})$$

Cuando la variable independiente es una variable dummy:

$$\begin{aligned} EMg_j &= P(y_i = 1|x_{i1}, x_{i2}, \dots, x_{ij} = 1, \dots, x_{iK}) - \\ &\quad P(y_i = 1|x_{i1}, x_{i2}, \dots, x_{ij} = 0, \dots, x_{iK}) \end{aligned}$$

Este tipo de modelos se basan en el modelo de Bernoulli, cuya función de densidad de probabilidad condicional y podemos planteárselo como:

$$\begin{aligned} P(y_i = 1|\mathbf{x}_i) &= p(x) \\ P(y_i = 0|\mathbf{x}_i) &= 1 - p(x) \\ Var(y_i|\mathbf{x}_i) &= p(x) \cdot (1 - p(x)) \end{aligned}$$

Algunos planteamientos adicionales

Cuando la variable dependiente es binaria (0 y 1) no se puede implementar una ecuación lineal de la forma tradicional, ya que no es posible determinar una ecuación:

$$y_i = \mathbf{x}_i \boldsymbol{\beta} + \varepsilon_i$$

Por otro lado, también enfrentaremos el mismo problema con modelos de probabilidad lineal:

$$P(y_i = 1|\mathbf{x}_i) = \mathbf{x}_i \boldsymbol{\beta} + \varepsilon_i$$

Y:

$$P(y_i = 0|\mathbf{x}_i) = \mathbf{x}_i \boldsymbol{\beta} + \varepsilon_i$$

En ambos casos es difícil lograr la estimación.

Modelos Logit y Probit

Este tipo de modelos suponen que tenemos que existe una variable latente que se puede expresar como una ecuación lineal dada por:

$$y_i^* = \mathbf{x}_i\boldsymbol{\beta} + \varepsilon_i$$

Donde ε_i es una variable aleatoria con función de densidad con media cero y distribución simétrica alrededor de cero. Dado lo anterior, para nosotros sólo es visible que:

$$y_i = \begin{cases} 1 & \text{si } y_i^* > 0 \\ 0 & \text{si } y_i^* \leq 0 \end{cases}$$

De esta forma tenemos una estructura de la probabilidad dada por:

$$\begin{aligned} P(y_i = 1 | \mathbf{x}_i) &= P(\mathbf{x}_i\boldsymbol{\beta} + \varepsilon_i > 0) = P(\varepsilon_i > -\mathbf{x}_i\boldsymbol{\beta}) = G(\mathbf{x}_i\boldsymbol{\beta}) \\ P(y_i = 0 | \mathbf{x}_i) &= P(\mathbf{x}_i\boldsymbol{\beta} + \varepsilon_i \leq 0) = P(\varepsilon_i \leq -\mathbf{x}_i\boldsymbol{\beta}) = 1 - G(\mathbf{x}_i\boldsymbol{\beta}) \end{aligned}$$

Donde \mathbf{x}_i es un vector de dimensión $K \times 1$ que contiene al menos el término constante y $\boldsymbol{\beta}$ es un vector de parámetros a estimar, de forma que asumiremos:

$$\mathbf{x}_i\boldsymbol{\beta} = \beta_1 + x_{i1}\beta_2 + \dots + x_{iK}\beta_K$$

Asumiremos que $G(\cdot)$ es una función de densidad acumulada de forma que:

$$0 < G(\mathbf{x}_i\boldsymbol{\beta}) < 1, \forall \mathbf{x}_i\boldsymbol{\beta} \in \mathbb{R}$$

En este caso utilizaremos dos modelos que dependen de la forma funcional de $G(\cdot)$ que está determinada por la distribución de ε_i . De esta forma tendremos dos modelos: Probit y Logit:

1. Modelo Probit, cuando

$$G(\mathbf{x}_i\boldsymbol{\beta}) = \Phi(\mathbf{x}_i\boldsymbol{\beta}) = \int_{-\infty}^{\mathbf{x}_i\boldsymbol{\beta}} \phi(\nu) d\nu$$

Donde $\phi(\cdot)$ es la densidad de probabilidad normal estándar:

$$\phi(\mathbf{x}_i\boldsymbol{\beta}) = \frac{1}{\sqrt{2\pi}} e^{-1/2\phi(\mathbf{x}_i\boldsymbol{\beta})}$$

2. Modelo Logit, cuando

$$G(\mathbf{x}_i\boldsymbol{\beta}) = \Lambda(\mathbf{x}_i\boldsymbol{\beta}) = \frac{e^{\mathbf{x}_i\boldsymbol{\beta}}}{1 + e^{\mathbf{x}_i\boldsymbol{\beta}}}$$

Donde $\lambda(\cdot)$ es la función de densidad acumulada logística.

Sin importar el modelo que estemos ocupando, la forma de interpretar el modelo es mediante el efecto marginal, cuando x_j sea una variable continua:

$$EMg_j = \frac{\partial}{\partial x_j} P(y = 1|\mathbf{x}_i) = \frac{\partial}{\partial x_j} G(\mathbf{x}_i\boldsymbol{\beta}) = g(\mathbf{x}_i\boldsymbol{\beta})\beta_j$$

Por su parte, cuando x_j sea una variable dicotómica:

$$\begin{aligned} EMg_j &= P(y = 1|\mathbf{x}_i, x_j = 1) - P(y = 1|\mathbf{x}_i, x_j = 0) \\ &= G(\mathbf{x}_i\boldsymbol{\beta}|x_j = 1) - G(\mathbf{x}_i\boldsymbol{\beta}|x_j = 0) \\ &= G(\beta_1 + x_2\beta_2 + \dots + \beta_j + \dots + x_K\beta_K) \\ &\quad - G(\beta_1 + x_2\beta_2 + \dots + 0 + \dots + x_K\beta_K) \end{aligned}$$

Estimación

Supongamos n observaciones de una variables aleatoria independientes e idénticamente distribuidas. En estos casos la función de densidad para cada y_i , $i = 1, 2, \dots, n$, estará dada por:

$$f(y_i|\mathbf{x}_i; \boldsymbol{\beta}) = [G(\mathbf{x}_i\boldsymbol{\beta})]^{y_i}[1 - G(\mathbf{x}_i\boldsymbol{\beta})]^{1-y_i}, \text{ donde } y_i = 0, 1$$

En estos términos la función de verosimilitud estará dada por:

$$\begin{aligned} L(\boldsymbol{\beta}) &= \prod_{i=1}^n f(y_i|\mathbf{x}_i; \boldsymbol{\beta}) \\ &= \prod_{i=1}^n [G(\mathbf{x}_i\boldsymbol{\beta})]^{y_i}[1 - G(\mathbf{x}_i\boldsymbol{\beta})]^{1-y_i} \end{aligned}$$

En versión logarítmica:

$$\begin{aligned} \ln L(\boldsymbol{\beta}) &= \sum_{i=1}^n \ln(f(y_i|\mathbf{x}_i; \boldsymbol{\beta})) \\ &= \sum_{i=1}^n y_i \ln[G(\mathbf{x}_i\boldsymbol{\beta})] + \sum_{i=1}^n (1 - y_i) \ln[1 - G(\mathbf{x}_i\boldsymbol{\beta})] \end{aligned}$$

Las condiciones de primer orden serán:

$$\frac{\partial}{\partial \beta} \ln L(\beta) = \sum_{i=1}^n y_i \frac{g(\mathbf{x}_i \beta)}{G(\mathbf{x}_i \beta)} \mathbf{x}'_i - \sum_{i=1}^n (1 - y_i) \frac{g(\mathbf{x}_i \beta)}{1 - G(\mathbf{x}_i \beta)} \mathbf{x}'_i$$

Finalmente, las condiciones de segundo orden y varianza estarán dada por:

$$\begin{aligned} \frac{\partial^2}{\partial \beta \partial \beta'} \ln L(\beta) &= - \sum_{i=1}^n \frac{y_i g(\mathbf{x}_i \beta)^2}{G(\mathbf{x}_i \beta)[1 - G(\mathbf{x}_i \beta)]} \mathbf{x}'_i \mathbf{x}_i \\ &= H(\beta) \end{aligned}$$

Por lo tanto:

$$\begin{aligned} Var(\hat{\beta}) &= [-\mathbb{E}[H(\beta)]]^{-1} \\ &= \left[\sum_{i=1}^n \frac{y_i g(\mathbf{x}_i \beta)^2}{G(\mathbf{x}_i \beta)[1 - G(\mathbf{x}_i \beta)]} \mathbf{x}'_i \mathbf{x}_i \right]^{-1} \end{aligned}$$

1.3.3. Modelos de respuesta multinomial y ordenada

En esta sección analizaremos modelos de respuesta discreta con más de 2 resultados posibles, los cuales se dividen:

1. Modelos de respuesta no ordenada, son modelos conocidos como de respuesta nominal, donde los valores de los distintos resultados son arbitrarios y no tienen un efecto en la estimación. Ejemplos de estos modelos son casos de selección de seguro, lugar de hospedaje, etc.
2. Modelos de respuesta ordenada, en estos modelos se asigna a cada un valor no arbitrario a la respuesta. Por ejemplo, modelos de calificación de crédito, modelos de preferencia de bienes, etc.

Modelos de respuesta multinomial

Logit multinomial

Este primer modelo aplica en situaciones en las que la unidad de respuesta o elección depende de las características individuales de los elementos de la muestra, pero no de la atribuciones de la elección.

Dado lo anterior, definiremos el modelo en términos de variables aleatorias que representan a la población subyacente. Sea y_i una variable aleatoria que toma valores en un conjunto $\{0, 1, 2, \dots, J\}$. Sea \mathbf{x}_i un conjunto de valores o regresores para $i = 1, 2, \dots, n$.

Como en el modelo de respuesta binaria, estimaremos el efecto de una variable en la probabilidad de ocurrencia de uno de los valores de respuesta. Dichas variables de respuesta estarán dados por:

$$P(y_i = j|\mathbf{x}_i), \text{ para } j = 1, 2, \dots, J$$

Dado que la suma de probabilidades debe ser 1, se suele tomar como práctica estimar $P(y_i = 0|\mathbf{x}_i)$ como diferencia del resto de los casos $j = \{1, 2, \dots, J\}$. Así, sea \mathbf{x}_i un vector de dimensión $1 \times K$. De esta forma, el modelo Logit tendrá como probabilidad de cada una de las respuestas a:

$$P(y_i = j|\mathbf{x}_i) = \frac{e^{\mathbf{x}_i \beta_j}}{1 + e^{\mathbf{x}_i \beta_1} + e^{\mathbf{x}_i \beta_2} + \dots + e^{\mathbf{x}_i \beta_J}}$$

Donde cada una de las β_j , $j = 1, \dots, J$ es de dimensión $K \times 1$. Es fácil observar que si la suma de probabilidades es 1, entonces, la probabilidad de $y = 0|\mathbf{x}_i$ estará dada por:

$$\begin{aligned} P(y_i = 0|\mathbf{x}_i) &= \frac{e^0}{1 + e^{\mathbf{x}_i \beta_1} + e^{\mathbf{x}_i \beta_2} + \dots + e^{\mathbf{x}_i \beta_J}} \\ &= \frac{1}{1 + e^{\mathbf{x}_i \beta_1} + e^{\mathbf{x}_i \beta_2} + \dots + e^{\mathbf{x}_i \beta_J}} \end{aligned}$$

De esta forma tenemos:

$$\begin{aligned} 1 &= \sum_{j=0}^J P(y_i = j|\mathbf{x}_i) \\ &= \frac{1}{1 + e^{\mathbf{x}_i \beta_1} + e^{\mathbf{x}_i \beta_2} + \dots + e^{\mathbf{x}_i \beta_J}} \\ &\quad + \frac{e^{\mathbf{x}_i \beta_1}}{1 + e^{\mathbf{x}_i \beta_1} + e^{\mathbf{x}_i \beta_2} + \dots + e^{\mathbf{x}_i \beta_J}} \\ &\quad + \dots + \\ &\quad + \frac{e^{\mathbf{x}_i \beta_J}}{1 + e^{\mathbf{x}_i \beta_1} + e^{\mathbf{x}_i \beta_2} + \dots + e^{\mathbf{x}_i \beta_J}} \end{aligned}$$

Para el proceso de estimación consideraremos la función de verosimilitud dada por:

$$L(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \dots, \boldsymbol{\beta}_J) = \prod_{i=1}^n \left[(P(j=0|\mathbf{x}_i))^{I(0)} \cdot (P(j=1|\mathbf{x}_i))^{I(1)} \cdots (P(j=J|\mathbf{x}_i))^{I(J)} \right]$$

Donde $I(j)$ es la función indicadora que toma el valor de 1 o 0 dependiendo de cual de las opciones posibles $j = 0, 1, 2, \dots, J$ es cierta para cada individuo $i = 1, 2, \dots, n$. De forma similar al caso de respuesta binaria en los modelos Logit Multinomiales se interpretan en sus efectos marginales y no de forma directa en los coeficientes $\boldsymbol{\beta}_j$.

Para una variable x_{ik} continua y para cada opción $j = 0, 1, 2, \dots, J$ el efecto marginal será:

$$\begin{aligned} EMg_k &= \frac{\partial P(y_i = j|\mathbf{x}_i)}{\partial x_{ik}} \\ &= P(y_i = j|\mathbf{x}_i) \cdot \left[\beta_{jk} - \frac{\sum_{h=1}^J \beta_{hk} \cdot e^{\mathbf{x}_i \boldsymbol{\beta}_h}}{1 + e^{\mathbf{x}_i \boldsymbol{\beta}_1} + e^{\mathbf{x}_i \boldsymbol{\beta}_2} + \dots + e^{\mathbf{x}_i \boldsymbol{\beta}_J}} \right] \end{aligned}$$

Donde β_{hk} es el k -ésimo elemento del vector $\boldsymbol{\beta}_h$. En el caso de que x_{ik} sea una variable dicotómica (o categórica) y para cada opción $j = 0, 1, 2, \dots, J$ el efecto marginal será:

$$EMg_k = P(y_i = j|\mathbf{x}_i, x_{ik} = 1) - P(y_i = j|\mathbf{x}_i, x_{ik} = 0)$$

Modelos de respuesta ordenada: Logit y Probit Ordinal

Sea y_i una variable que representa ordenada que toma los valores de $\{0, 1, 2, \dots, J\}$. En estos casos en que importa el orden de la respuesta existen dos mecanismos de estimación: el Modelo Probit y el Modelo Logit.

Modelo Probit Ordinal. Este modelo se puede derivar de forma similar al modelo probit de dos respuestas 0 y 1, probit binario. Así, partimos de una variable latente y_i^* definida como:

$$y_i^* = \mathbf{x}_i \boldsymbol{\beta} + \boldsymbol{\varepsilon}_i; \quad \boldsymbol{\varepsilon}_i \sim N(0, 1)$$

Donde $\boldsymbol{\beta}$ es un vector $(K - 1) \times 1$ ya que en este caso asumiremos que no se considera un término constante, por las razones que mostramos más

adelante. Sean $\alpha_1, \alpha_2, \dots, \alpha_J$ un conjunto de umbrales que partitionan la probabilidad de la siguiente forma:

$$\begin{aligned} y_i = 0 & \quad \text{si} \quad y_i^* \leq \alpha_1 \\ y_i = 1 & \quad \text{si} \quad \alpha_1 < y_i^* \leq \alpha_2 \\ & \vdots \\ y_i = j & \quad \text{si} \quad \alpha_j < y_i^* \leq \alpha_{j+1} \\ & \vdots \\ y_i = J & \quad \text{si} \quad \alpha_J < y_i^* \end{aligned}$$

Dado que ε_i tiene una distribución normal estándar, para cada uno de los casos anteriores podemos establecer:

$$\begin{aligned} P(y_i = 0 | \mathbf{x}_i) &= P(y_i^* \leq \alpha_1 | \mathbf{x}_i) = P(\mathbf{x}_i \boldsymbol{\beta} + \varepsilon_i \leq \alpha_1 | \mathbf{x}_i) \\ P(y_i = 1 | \mathbf{x}_i) &= P(\alpha_1 < y_i^* \leq \alpha_2 | \mathbf{x}_i) = P(\alpha_1 < \mathbf{x}_i \boldsymbol{\beta} + \varepsilon_i \leq \alpha_2 | \mathbf{x}_i) \\ & \vdots \\ P(y_i = j | \mathbf{x}_i) &= P(\alpha_j < y_i^* \leq \alpha_{j+1} | \mathbf{x}_i) = P(\alpha_j < \mathbf{x}_i \boldsymbol{\beta} + \varepsilon_i \leq \alpha_{j+1} | \mathbf{x}_i) \\ & \vdots \\ P(y_i = J | \mathbf{x}_i) &= P(\alpha_J < y_i^* | \mathbf{x}_i) = P(\alpha_J < \mathbf{x}_i \boldsymbol{\beta} + \varepsilon_i | \mathbf{x}_i) \end{aligned}$$

Desarrollando las ecuaciones anteriores tenemos para el caso de $y_i = 0$:

$$\begin{aligned} P(y_i = 0 | \mathbf{x}_i) &= P(\mathbf{x}_i \boldsymbol{\beta} + \varepsilon_i \leq \alpha_1 | \mathbf{x}_i) \\ &= P(\varepsilon_i \leq \alpha_1 - \mathbf{x}_i \boldsymbol{\beta} | \mathbf{x}_i) \\ &= \Phi(\alpha_1 - \mathbf{x}_i \boldsymbol{\beta}) \end{aligned}$$

En el caso de $y_i = j$, $j = 1, 2, \dots, J-1$:

$$\begin{aligned} P(y_i = j | \mathbf{x}_i) &= P(\alpha_j < \mathbf{x}_i \boldsymbol{\beta} + \varepsilon_i \leq \alpha_{j+1} | \mathbf{x}_i) \\ &= P(\alpha_j - \mathbf{x}_i \boldsymbol{\beta} < \varepsilon_i \leq \alpha_{j+1} - \mathbf{x}_i \boldsymbol{\beta} | \mathbf{x}_i) \\ &= P(\varepsilon_i \leq \alpha_{j+1} - \mathbf{x}_i \boldsymbol{\beta} | \mathbf{x}_i) - P(\varepsilon_i < \alpha_j - \mathbf{x}_i \boldsymbol{\beta} | \mathbf{x}_i) \\ &= \Phi(\alpha_{j+1} - \mathbf{x}_i \boldsymbol{\beta}) - \Phi(\alpha_j - \mathbf{x}_i \boldsymbol{\beta}) \end{aligned}$$

Finalmente, para el caso $y_i = J$:

$$\begin{aligned}
P(y_i = J | \mathbf{x}_i) &= P(\alpha_J < \mathbf{x}_i \boldsymbol{\beta} + \boldsymbol{\varepsilon}_i | \mathbf{x}_i) \\
&= P(\alpha_J - \mathbf{x}_i \boldsymbol{\beta} < \boldsymbol{\varepsilon}_i | \mathbf{x}_i) \\
&= 1 - P(\boldsymbol{\varepsilon}_i < \alpha_J - \mathbf{x}_i \boldsymbol{\beta} | \mathbf{x}_i) \\
&= 1 - \Phi(\alpha_J - \mathbf{x}_i \boldsymbol{\beta})
\end{aligned}$$

Es sencillo observar y analizar que:

$$1 = \sum_{j=0}^J P(y_i = j | \mathbf{x}_i)$$

Notemos que en este modelo los cambios en la probabilidad no serán determinados por \mathbf{x}_i , ya que lo que determina el cambio es el umbral α_j , $j = 0, 1, \dots, J$. También en este caso los coeficientes no se interpretan de forma directa, en cambio se utiliza el efecto marginal que tiene expresiones similares al del probit bivariado como más adelante mostraremos.

Modelo Logit Ordinal. Este modelo se puede derivar de forma similar al modelo Probit Ordinal con una variable latente y_i^* , también con $\boldsymbol{\beta}$ es un vector $(K - 1) \times 1$ ya que en este caso asumiremos que no se considera un término constante, por las razones que hemos mostrado anteriormente. Igualmente, sean $\alpha_1, \alpha_2, \dots, \alpha_J$ un conjunto de umbrales que partitionan la probabilidad considerando una función logística $\Lambda(\cdot)$ de la siguiente forma:

$$\boldsymbol{\varepsilon}_i \sim \Lambda(\cdot)$$

$$P(y_i = J | \mathbf{x}_i) = P(\alpha_J < y_i^* | \mathbf{x}_i) = P(\alpha_J < \mathbf{x}_i \boldsymbol{\beta} + \boldsymbol{\varepsilon}_i | \mathbf{x}_i)$$

Desarrollando las ecuaciones de probabilidad de forma similar al Probit Multinomial tenemos para el caso de $y_i = 0$:

$$\begin{aligned}
P(y_i = 0 | \mathbf{x}_i) &= P(y_i^* \leq \alpha_1 | \mathbf{x}_i) \\
&= P(\mathbf{x}_i \boldsymbol{\beta} + \boldsymbol{\varepsilon}_i \leq \alpha_1 | \mathbf{x}_i) \\
&= P(\boldsymbol{\varepsilon}_i \leq \alpha_1 - \mathbf{x}_i \boldsymbol{\beta} | \mathbf{x}_i) \\
&= \Lambda(\alpha_1 - \mathbf{x}_i \boldsymbol{\beta})
\end{aligned}$$

En el caso de $y_i = j$, $j = 1, 2, \dots, J - 1$:

$$\begin{aligned}
P(y_i = j | \mathbf{x}_i) &= P(\alpha_j < y_i^* \leq \alpha_{j+1} | \mathbf{x}_i) \\
&= P(\alpha_j < \mathbf{x}_i \boldsymbol{\beta} + \boldsymbol{\varepsilon}_i \leq \alpha_{j+1} | \mathbf{x}_i) \\
&= P(\alpha_j - \mathbf{x}_i \boldsymbol{\beta} < \boldsymbol{\varepsilon}_i \leq \alpha_{j+1} - \mathbf{x}_i \boldsymbol{\beta} | \mathbf{x}_i) \\
&= P(\boldsymbol{\varepsilon}_i \leq \alpha_{j+1} - \mathbf{x}_i \boldsymbol{\beta} | \mathbf{x}_i) - (P(\boldsymbol{\varepsilon}_i < \alpha_j - \mathbf{x}_i \boldsymbol{\beta} | \mathbf{x}_i)) \\
&= \Lambda(\alpha_{j+1} - \mathbf{x}_i \boldsymbol{\beta}) - \Lambda(\alpha_j - \mathbf{x}_i \boldsymbol{\beta})
\end{aligned}$$

Finalmente, para el caso $y_i = J$:

$$\begin{aligned}
P(y_i = J | \mathbf{x}_i) &= P(\alpha_J < y_i^* | \mathbf{x}_i) \\
&= P(\alpha_J < \mathbf{x}_i \boldsymbol{\beta} + \boldsymbol{\varepsilon}_i | \mathbf{x}_i) \\
&= P(\alpha_J - \mathbf{x}_i \boldsymbol{\beta} < \boldsymbol{\varepsilon}_i | \mathbf{x}_i) \\
&= 1 - P(\boldsymbol{\varepsilon}_i < \alpha_J - \mathbf{x}_i \boldsymbol{\beta} | \mathbf{x}_i) \\
&= 1 - \Lambda(\alpha_J - \mathbf{x}_i \boldsymbol{\beta})
\end{aligned}$$

También, es sencillo observar y analizar que en los Logit Multinomiales:

$$1 = \sum_{j=0}^J P(y_i = j | \mathbf{x}_i)$$

Notemos que en este modelo los cambios en la probabilidad no serán determinados por \mathbf{x}_i , ya que lo que determina el cambio es el umbral α_j , $j = 0, 1, \dots, J$. También en este caso los coeficientes no se interpretan de forma directa, en cambio se utiliza el efecto marginal que tiene expresiones similares al del probit bivariado como más adelante mostraremos.

Estimación de modelos Probit y Logit Ordinal. Para la estimación implementaremos un proceso de maximización de la siguiente función de verosimilitud:

$$L(\alpha, \boldsymbol{\beta}) = \prod_{i=1}^n P(y_i = 0 | \mathbf{x}_i)^{I(0)} \cdot P(y_i = 1 | \mathbf{x}_i)^{I(1)} \cdots P(y_i = J | \mathbf{x}_i)^{I(J)}$$

Donde $I(j)$ es la función indicadora que toma el valor de 0 y 1 si el individuo i-ésimo tiene como respuesta alguno de los posibles valores de $j = 0, 1, \dots, J$. La función es válida sin importar la función $\Phi(\cdot)$ o $\Lambda(\cdot)$ que utilicemos para determinar la $P(y_i = j | \mathbf{x}_i)$, para $j = 1, 2, \dots, J$. Por lo cual simbolizaremos indistintamente estas funciones como $G(\cdot)$.



Figura 1.14: División del conjunto de datos, retomado de Hastie, Tibshirani, y Friedman (2017, p. 222) [HTF17]

Finalmente, sin importar el modelo que estemos ocupando, Probit o Logit ordinal, la forma de interpretar el modelo es mediante el efecto marginal, cuando x_k sea una variable continua:

$$EMg_k = \frac{\partial}{\partial x_k} P(y_i = j | \mathbf{x}_i) = \frac{\partial}{\partial x_k} G(\alpha_j - \mathbf{x}_i \boldsymbol{\beta}) = -\beta_k \cdot g(\alpha_j - \mathbf{x}_i \boldsymbol{\beta})$$

Por su parte, cuando x_j sea una variable dicotómica:

$$\begin{aligned} EMg_k &= P(y_i = j | \mathbf{x}_i, x_{ik} = 1) - P(y_i = j | \mathbf{x}_i, x_{ik} = 0) \\ &= G(\alpha_j - \mathbf{x}_i \boldsymbol{\beta} | \mathbf{x}_i, x_{ik} = 1) - G(\alpha_j - \mathbf{x}_i \boldsymbol{\beta} | \mathbf{x}_i, x_{ik} = 0) \\ &= G(\alpha_j - (\beta_1 + x_2\beta_2 + \dots + \beta_k + \dots + x_K\beta_K)) \\ &\quad - G(\alpha_j - (\beta_1 + x_2\beta_2 + \dots + 0 + \dots + x_K\beta_K)) \end{aligned}$$

También es posible hacer un efecto marginal intra respuestas, es decir, para cambios entre una respuesta j y una $j + h$, la cual será:

$$EMg_{j-h} = \beta_k \cdot [g(\alpha_h - \mathbf{x}_i \boldsymbol{\beta}) - g(\alpha_j - \mathbf{x}_i \boldsymbol{\beta})], \text{ con } h < j$$

Donde $k = 1, 2, \dots, K - 1$ y $j = 0, 1, \dots, J$.

Ajuste y separación del conjunto de datos

De forma similar al caso de regresión, dividiremos al conjunto de datos conforme se describe en la Figura 1.14.

Evaluando los modelos de aprendizaje supervisado

Matriz de confusión

Una vez que se entrena un problema de aprendizaje automatizado supervisado en un conjunto de datos históricos, se prueba el modelo obtenido mediante el uso de datos del conjunto de entrenamiento. De esta forma, es

possible comparar las predicciones del modelo entrenado con los valores reales de la variable sujeta de análisis. La matriz de confusión proporciona un medio para evaluar el éxito de un problema de clasificación y dónde se cometan errores (es decir, dónde se vuelve 'confuso').

En el Cuadro 1.1 se muestra un ejemplo de la forma en que se suele mostrar una matriz de confusión.

		Predicciones	
		Positivas (1)	Negativas (0)
Real	Positivas (1)	True Positive (TP)	False Negative (FN)
	Negativas (0)	False Positive (FP)	True Negative (TN)

Cuadro 1.1: Matriz de Confusión

Con base en la matriz de confusión en el Cuadro 1.1, se pueden construir métricas que se calculan de la siguiente manera:

$$Precision = \frac{TP}{TP + FP} \quad (1.45)$$

$$Recuperacion = \frac{TP}{TP + FN} \quad (1.46)$$

$$F1 = 2 \times \frac{Precision \times Recuperacion}{Precision + Recuperacion} \quad (1.47)$$

La medida $F1$ o $F1 - score$ combina precisión y recuperación mediante la media armónica de los dos valores. De esta forma, siempre se ubicará entre ambos indicadores.

1.3.4. 4.2.2 Regresión logística para el estudio de equidad de violencia.

Ver "05_Ejemplo_Logit_Delitos" en GitHub.

1.3.5. 4.2.1 Regresión logística con información de videojuegos.

Planteamiento del caso

Supuestos y necesidades del caso:

1. Determinar cuáles son las features/características/steam tags que se asocian con videojuegos para PC existentes.
2. Hay varias tiendas de videojuegos (Steam, itch, etc.) que utilizan estas tags para clasificar a sus juegos.

Propuesta: Usar un modelo de respuesta ordenada

Sea y una variable que representa una respuesta ordenada que toma valores $\{0, 1, 2, \dots, J\} \in \mathbb{Z}$. El modelo de respuesta ordenada (conocido como Probit o Logit ordenado) para y (condicional en un vector de variables explicativas \mathbf{X} -features, características, tags, etc.) es derivado a partir de un modelo de variable latente y^* , el cual se puede escribir como:

$$y^* = \mathbf{X}\boldsymbol{\beta} + \varepsilon, \quad \varepsilon | \mathbf{X} \sim f(\varepsilon) \quad (1.48)$$

Donde $f(\cdot)$ es una función de densidad de probabilidad simétrica cuya función de densidad acumulada será denotada por $F(\cdot)$, $\boldsymbol{\beta}$ es un vector de dimensión $K \times 1$ y, por razones que más adelante explicamos, \mathbf{X} no tiene término constante. Por otro lado, sean $\mu_1 < \mu_2 < \dots < \mu_J$ puntos de corte o parámetros umbrales que definen lo siguiente:

$$\begin{aligned} y &= 0 && \text{si } y^* \leq \mu_1 \\ y &= 1 && \text{si } \mu_1 < y^* \leq \mu_2 \\ &&& \vdots \\ y &= J && \text{si } y^* > \mu_J \end{aligned} \quad (1.49)$$

De esta forma y considerando el conjunto de desigualdades en la ecuación (1.49) y la ecuación (1.48), podemos determinar las siguientes probabilidades a partir de una partición de una función de distribución dada:

$$\begin{aligned} \mathbf{P}(y = 0 | \mathbf{X}) &= \mathbf{P}(y^* \leq \mu_1 | \mathbf{X}) \\ &= \mathbf{P}(\mathbf{X}\boldsymbol{\beta} + \varepsilon \leq \mu_1 | \mathbf{X}) \\ &= \mathbf{P}(\varepsilon \leq \mu_1 - \mathbf{X}\boldsymbol{\beta} | \mathbf{X}) \\ &= F(\mu_1 - \mathbf{X}\boldsymbol{\beta}) \end{aligned}$$

$$\begin{aligned}
\mathbf{P}(y = 1 | \mathbf{X}) &= \mathbf{P}(\mu_1 < y^* \leq \mu_2 | \mathbf{X}) \\
&= \mathbf{P}(\mu_1 < \mathbf{X}\boldsymbol{\beta} + \varepsilon \leq \mu_2 | \mathbf{X}) \\
&= \mathbf{P}(\mu_1 - \mathbf{X}\boldsymbol{\beta} < \varepsilon \leq \mu_2 - \mathbf{X}\boldsymbol{\beta} | \mathbf{X}) \\
&= F(\mu_2 - \mathbf{X}\boldsymbol{\beta}) - F(\mu_1 - \mathbf{X}\boldsymbol{\beta})
\end{aligned}$$

⋮

$$\begin{aligned}
\mathbf{P}(y = J-1 | \mathbf{X}) &= \mathbf{P}(\mu_{J-1} < y^* \leq \mu_J | \mathbf{X}) \\
&= \mathbf{P}(\mu_{J-1} < \mathbf{X}\boldsymbol{\beta} + \varepsilon \leq \mu_J | \mathbf{X}) \\
&= \mathbf{P}(\mu_{J-1} - \mathbf{X}\boldsymbol{\beta} < \varepsilon \leq \mu_J - \mathbf{X}\boldsymbol{\beta} | \mathbf{X}) \\
&= F(\mu_J - \mathbf{X}\boldsymbol{\beta}) - F(\mu_{J-1} - \mathbf{X}\boldsymbol{\beta})
\end{aligned}$$

$$\begin{aligned}
\mathbf{P}(y = J | \mathbf{X}) &= \mathbf{P}(y^* > \mu_J | \mathbf{X}) \\
&= \mathbf{P}(\mathbf{X}\boldsymbol{\beta} + \varepsilon > \mu_J | \mathbf{X}) \\
&= \mathbf{P}(\varepsilon > \mu_J - \mathbf{X}\boldsymbol{\beta} | \mathbf{X}) \\
&= 1 - F(\mu_J - \mathbf{X}\boldsymbol{\beta})
\end{aligned}$$

Note que en este caso asumimos que \mathbf{X} no tiene término constante, ya que la incorporamos mediante los umbrales μ_j , $j = 1, 2, \dots, J$. Un caso particular de esta modelación son los modelos de clasificación binarios en los que $J = 1$. En la figura 1.15 ilustramos la partición de la función de densidad $f(\cdot)$ que resultaría de las ecuaciones anteriores.

De esta manera, los resultados de la estimación se pueden interpretar a través de sus efectos marginales dados por:

$$\frac{\partial \mathbf{P}(y = j | \mathbf{X})}{\partial x_k} = -\beta_k(f(\mu_j - \mathbf{X}\boldsymbol{\beta}) - f(\mu_{j-1} - \mathbf{X}\boldsymbol{\beta})) \text{ para } 0 < j < J \quad (1.50)$$

Así, este efecto se interpreta como la contribución que tiene la variable x_k a la probabilidad de que la variable y tome el valor de j .

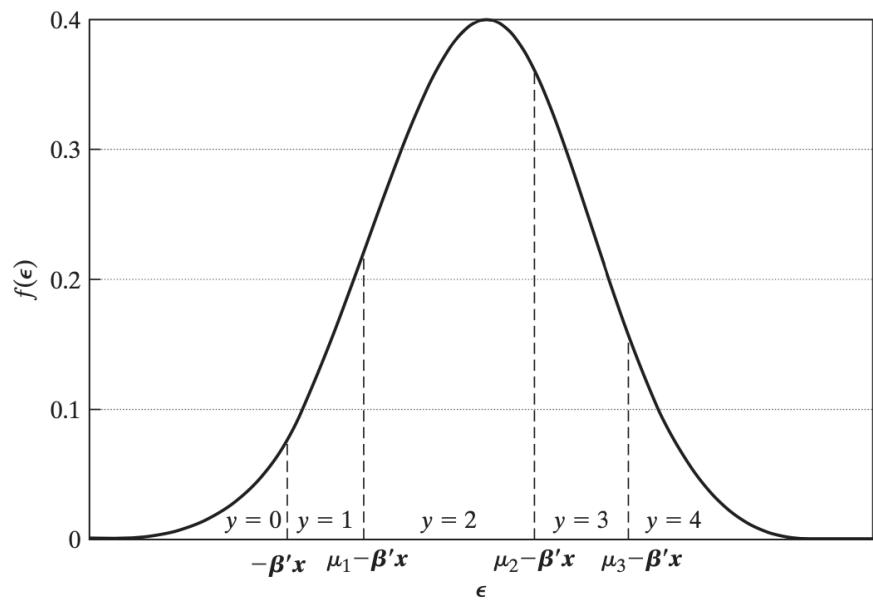


Figura 1.15: Probabilidades en un modelo de respuesta ordenada, considerando $J = 4$ (retomado de Greene (2012, 788) [Gre12])

Aplicación al caso de videojuegos

En el caso de los videojuegos, podemos aplicar el modelo de respuesta ordenada asumiendo algunas cosas. Primero, F tendrá una forma de función logística, para facilitar el proceso computacional. Segundo, la variable y será particionada en rangos de ingresos obtenidos por los videojuegos en su primer año (y^*)—año de lanzamiento—, por ejemplo:

$$y^* = \mathbf{X}\boldsymbol{\beta} + \varepsilon, \quad \varepsilon | \mathbf{X} \sim f(\varepsilon)$$

Where y^* is the revenue per year, \mathbf{X} are the characteristics and other variables of video games, $\boldsymbol{\beta}$ are the coefficients and ε is the error term.

$$\begin{aligned} y &= 0 && \text{if } y^* \leq 1 \text{M USD} \\ y &= 1 && \text{if } 1 \text{ MM USD} < y^* \leq 5 \text{ MM USD} \\ y &= 2 && \text{if } 5 \text{ MM USD} < y^* \leq 20 \text{ MM USD} \\ y &= 3 && \text{if } y^* > 20 \text{ MM USD} \end{aligned}$$

$$\begin{aligned} \mathbf{P}(y = 0 | \mathbf{X}) &= \mathbf{P}(y^* \leq \mu_1 | \mathbf{X}) \\ &= \mathbf{P}(\mathbf{X}\boldsymbol{\beta} + \varepsilon \leq \mu_1 | \mathbf{X}) \\ &= \mathbf{P}(\varepsilon \leq \mu_1 - \mathbf{X}\boldsymbol{\beta} | \mathbf{X}) \\ &= F(\mu_1 - \mathbf{X}\boldsymbol{\beta}) \end{aligned}$$

$$\begin{aligned} \mathbf{P}(y = 1 | \mathbf{X}) &= \mathbf{P}(\mu_1 < y^* \leq \mu_2 | \mathbf{X}) \\ &= \mathbf{P}(\mu_1 < \mathbf{X}\boldsymbol{\beta} + \varepsilon \leq \mu_2 | \mathbf{X}) \\ &= \mathbf{P}(\mu_1 - \mathbf{X}\boldsymbol{\beta} < \varepsilon \leq \mu_2 - \mathbf{X}\boldsymbol{\beta} | \mathbf{X}) \\ &= F(\mu_2 - \mathbf{X}\boldsymbol{\beta}) - F(\mu_1 - \mathbf{X}\boldsymbol{\beta}) \end{aligned}$$

$$\begin{aligned} \mathbf{P}(y = 2 | \mathbf{X}) &= \mathbf{P}(\mu_2 < y^* \leq \mu_3 | \mathbf{X}) \\ &= \mathbf{P}(\mu_2 < \mathbf{X}\boldsymbol{\beta} + \varepsilon \leq \mu_3 | \mathbf{X}) \\ &= \mathbf{P}(\mu_2 - \mathbf{X}\boldsymbol{\beta} < \varepsilon \leq \mu_3 - \mathbf{X}\boldsymbol{\beta} | \mathbf{X}) \\ &= F(\mu_3 - \mathbf{X}\boldsymbol{\beta}) - F(\mu_2 - \mathbf{X}\boldsymbol{\beta}) \end{aligned}$$

$$\begin{aligned}
\mathbf{P}(y = 3|\mathbf{X}) &= \mathbf{P}(y^* > \mu_3|\mathbf{X}) \\
&= \mathbf{P}(\mathbf{X}\beta + \varepsilon > \mu_3|\mathbf{X}) \\
&= \mathbf{P}(\varepsilon > \mu_3 - \mathbf{X}\beta|\mathbf{X}) \\
&= 1 - F(\mu_3 - \mathbf{X}\beta)
\end{aligned}$$

$$\mathbf{P}(y = 0|\mathbf{X}) = \mu_1 - \mathbf{X}\beta$$

$$\mathbf{P}(y = 1|\mathbf{X}) = \mu_2 - \mathbf{X}\beta$$

$$\mathbf{P}(y = 2|\mathbf{X}) = \mu_3 - \mathbf{X}\beta$$

$$\varepsilon$$

Supongamos que tenemos n observaciones o, en este caso, videojuegos, los cuales son indexados con $i = 1, 2, 3, \dots, n$. Así, indexaremos a la variable de respuesta y como y_i , por lo que diremos que buscamos estimar las siguientes probabilidades:

$$\begin{aligned}
\mathbf{P}(\text{Revenue}_i \leq 1\text{M USD}|\mathbf{X}_i) &= \mathbf{P}(\varepsilon_i \leq \mu_1 - \mathbf{X}_i\beta|\mathbf{X}_i) \\
&= F(\mu_1 - \mathbf{X}_i\beta)
\end{aligned}$$

$$\begin{aligned}
\mathbf{P}(1 \text{ MM USD} < \text{Revenue}_i \leq 5 \text{ MM USD}|\mathbf{X}_i) &= \mathbf{P}(\mu_1 - \mathbf{X}_i\beta < \varepsilon_i \leq \mu_2 - \mathbf{X}_i\beta|\mathbf{X}_i) \\
&= F(\mu_2 - \mathbf{X}_i\beta) - F(\mu_1 - \mathbf{X}_i\beta)
\end{aligned}$$

$$\begin{aligned}
\mathbf{P}(5 \text{ MM USD} < \text{Revenue}_i \leq 20 \text{ MM USD}|\mathbf{X}_i) &= \mathbf{P}(\mu_2 - \mathbf{X}_i\beta < \varepsilon_i \leq \mu_3 - \mathbf{X}_i\beta|\mathbf{X}_i) \\
&= F(\mu_3 - \mathbf{X}_i\beta) - F(\mu_2 - \mathbf{X}_i\beta)
\end{aligned}$$

$$\begin{aligned}\mathbf{P}(\text{Revenue}_i > 20 \text{ MM USD} | \mathbf{X}_i) &= \mathbf{P}(\varepsilon_i > \mu_3 - \mathbf{X}_i \boldsymbol{\beta} | \mathbf{X}_i) \\ &= 1 - F(\mu_3 - \mathbf{X}_i \boldsymbol{\beta})\end{aligned}$$

Donde, particularmente, F es una función logistica. Nuestro objetivo es maximizar la función de verosimilitud (función que maximiza el valor de los parámetros μ_1 , μ_2 , μ_3 y $\boldsymbol{\beta}$ dada la información disponible):

$$\begin{aligned}\mathbf{L}(\boldsymbol{\theta} | \text{Data}) &= \prod_{y_i=0} \mathbf{P}(\text{Revenue}_i \leq 1 \text{ M USD} | \mathbf{X}_i) \\ &\quad \times \prod_{y_i=1} \mathbf{P}(1 \text{ MM USD} < \text{Revenue}_i \leq 5 \text{ MM USD} | \mathbf{X}_i) \\ &\quad \times \prod_{y_i=2} \mathbf{P}(5 \text{ MM USD} < \text{Revenue}_i \leq 20 \text{ MM USD} | \mathbf{X}_i) \\ &\quad \times \prod_{y_i=3} \mathbf{P}(\text{Revenue}_i > 20 \text{ MM USD} | \mathbf{X}_i) \\ &= \prod_{y_i=0} (F(\mu_1 - \mathbf{X}_i \boldsymbol{\beta})) \times \prod_{y_i=1} (F(\mu_2 - \mathbf{X}_i \boldsymbol{\beta}) - F(\mu_1 - \mathbf{X}_i \boldsymbol{\beta})) \\ &\quad \times \prod_{y_i=2} (F(\mu_3 - \mathbf{X}_i \boldsymbol{\beta}) - F(\mu_2 - \mathbf{X}_i \boldsymbol{\beta})) \times \prod_{y_i=3} (1 - F(\mu_3 - \mathbf{X}_i \boldsymbol{\beta}))\end{aligned}$$

where:

$$\boldsymbol{\theta} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \boldsymbol{\beta} \end{bmatrix}$$

Así, el problema consiste en resolver:

$$\begin{aligned}\max_{\boldsymbol{\theta}} \mathbf{L}(\boldsymbol{\theta} | \text{Data}) &= \prod_{\text{Revenue}_i \leq 1 \text{ M USD}} (F(\mu_1 - \mathbf{X}_i \boldsymbol{\beta})) \\ &\quad \times \prod_{1 \text{ MM USD} < \text{Revenue}_i \leq 5 \text{ MM USD} | \mathbf{X}_i} (F(\mu_2 - \mathbf{X}_i \boldsymbol{\beta}) - F(\mu_1 - \mathbf{X}_i \boldsymbol{\beta})) \\ &\quad \times \prod_{5 \text{ MM USD} < \text{Revenue}_i \leq 20 \text{ MM USD}} (F(\mu_3 - \mathbf{X}_i \boldsymbol{\beta}) - F(\mu_2 - \mathbf{X}_i \boldsymbol{\beta})) \\ &\quad \times \prod_{\text{Revenue}_i > 20 \text{ MM USD}} (1 - F(\mu_3 - \mathbf{X}_i \boldsymbol{\beta}))\end{aligned}$$

Tags	Juegos	Tags	Juegos	Tags	Juegos	Tags	Juegos	Tags	Juegos	Tags	Juegos
Indie	36,782	Combat	3,866	Multiple Endings	2,065	Historical	1,313	Musica	910	Demons	626
Action	25,650	Female Protagonist	3,857	Roguelike	2,054	Walking Simulator	1,312	Competitive	882	Destrucción	616
Adventure	23,680	Violent	3,798	Puzzle Platformer	2,047	Resource Management	1,303	Dark Humor	881	Turn-Based RPG	616
Casual	23,418		0	Shooter	1,394	Interactive Fiction	1,293	Aliens	880	Party-Based RPG	594
Singleplayer	23,397	Open World	3,604	Shoot Em Up	1,933	Memes	1,252	Clicker	872	Real-Time	589
2D	12,837	Sexual Content	3,598	Linear	1,849	Dungeon Crawler	1,240	Nature	866	2D Fighter	583
Strategy	11,854	Visual Novel	3,502	Character Customization	1,813	Score Attack	1,229	Driving	862	Automobile Sim	582
Simulation	11,800	Horror	3,500	Roleplay-Down	1,805	Space Sim	1,220	Industrial	860	Robotics 3	581
RPG	11,097	Turn-Based	3,359	Old School	1,794	Surreal	1,211	Fights	857	Rhythm	571
Puzzle	9,354	Action-Adventure	3,342	Massively Multiplayer	1,738	Top-Down Shooter	1,212	Arena Shooter	846	Modifiable	565
Multiplayer	8,185	Co-op	3,295	Procedural Generation	1,738	Experimental	1,204	Detective	846	Nonlinear	553
Atmospheric	8,117	Top-Down	3,289	Action RPG	1,735	Post-apocalyptic	1,201	Party	845	Modern	551
Early Access	7,609	Management	3,282	Board-Based Strategy	1,734	Stealth	1,198	Robots	837	Horror	543
Platformer	7,103	Third Person	3,240	Replay Value	1,706	RTS	1,167	Plasma Death	836	Dystopian	541
Story Rich	6,711	Cartoon	3,210	Classic	1,672	Isometric	1,151	Immersive Sim	824	Parkour	539
3D	6,584	Gore	3,205	Local Co-Op	1,651	Stealth	1,146	Investigation	820	Comic Book	534
Shooter	6,381	FPS	3,194	Horror	1,641	Dark Fantasy	1,129	Precision Platformer	815	Thriller	534
Pixel Graphics	6,157	Sports	3,194	Horror	1,637	Horror	1,128	Robotics	809	Alternate History	515
Fantasy	5,910	Mystery	2,704	Hidden Object	1,597	Narration	1,125	Battlem' up	805	Artificial Intelligence	496
Colorful	5,629	Point & Click	2,671	Logic	1,586	1990's	1,112	NSFW	781	CRPG	490
Arcade	5,562	Psychological	2,660	Mature	1,582	Drama	1,104	Time Management	782	Grand Strategy	489
Horror	5,504	Stylized	2,659	Magi	1,579	Top-Down	1,099	Board-Based	767	Cryptic	487
Anime	5,425	Science	2,531	Horror	1,558	Base Building	1,095	Universal	726	Space Sim	483
Cute	5,417	Minimalist	2,591	3D Platformer	1,519	Card Game	1,077	Metrovania	727	Lore-Rich	471
First-Person	5,351	Physics	2,573	Futuristic	1,518	Abstract	1,072	Team-Based	718	Mouse only	468
Funny	5,160	PvP	2,498	Fast-Paced	1,480	Hental	1,051	City Builder	702	Supernatural	463
Soundtrack	4,866	Violent	2,464	Turn-Based Combat	1,467	Horror	1,030	Real-Time Tactics	690	MMORPG	454
VR	4,862	Building	2,456	Horror	1,466	Third-Person Shooter	1,029	Strategy RPG	677	Addictive	451
Exploration	4,806	Sandbox	2,415	Crafting	1,451	4 Player Local	1,005	Loot	672	Souls-like	448
Retro	4,734	Cartoony	2,389	Bullet Hell	1,447	Tower Defense	997	Psychedelic	669	Trading	448
Sci-fi	4,601	2D Platformer	2,378	Short	1,428	2.5D	992	Character Action Game	666	Horror	446
Great Soundtrack	4,589	Psychological Horror	2,323	Medieval	1,424	Cyberpunk	986	Third-Person Shooter	662	Split Screen	436
Free to Play	4,473	Horror	2,317	Horror	1,417	Military	979	Conversation	657	Roleplay	431
Difficult	4,445	Side Scroller	2,243	Romance	1,373	Robots	963	Tabletop	653	Open World Survival Craft	429
Survival	4,410	Realistic	2,220	Hack and Slash	1,365	Action Roguelike	942	Level Editor	651	Science	426
Family Friendly	4,231	Racing	2,198	Choose Your Own Adventure	1,325	Immersive	934	Runner	647	Mythology	425
Dark	3,940	Controller	2,148	Turn-Based Tactics	1,325	Board Game	916	Life Sim	645	Inventory Management	413
Relaxing	3,924	Local Multiplayer	2,136	RPGMaker	1,317	Wargame	936	Philosophical	613		

There is a large quantity of tags found in the database, thus it makes complicated to classify a game into a single category. However, tag's value lies in the fact that users are the ones who tag a game. It tells us how a user perceives a game.

There are 417 tags in the database with a particular category for each one in which a game can be.

Figura 1.16: En la tabla de datos ubicamos 417 tags distintas

Datos

Usamos como fuente: <https://games-stats.com/>, una descarga de datos de 2021. La tabla de datos originalmente descargada contenía información de 53,650 videojuegos para computadora de los últimos 40 años, acumulando 18.5 billones de dólares.

La tabla de datos de games-stats.com se utilizó como base de datos central debido a su tamaño y variables disponibles. Fue validado con otros datos encontrados en línea, y tenían un tamaño similar entre ellos. Se agregaron más variables con el propósito de enriquecer el análisis.

Hay una gran cantidad de etiquetas (tags) encontradas en la tabla de datos, por lo que resulta complicado clasificar un juego en una sola categoría. Sin embargo, el valor de la etiqueta radica en el hecho de que son los usuarios quienes etiquetan un juego. Esto nos dice cómo un usuario percibe un juego. En total ubicamos 417 tags distintas.

En este caso, requerimos una reducción de dimensionalidad. El modelo debía evaluar más de 450 etiquetas diferentes, algo inviable. Tener 450 etiquetas para evaluar reduce la simplicidad e interpretabilidad del modelo. Así, aplicamos un par de metodologías:

- Extreme Random Decision Tree method: Que es una metodología de Machine Learning que ayudará a consider menos variables dada la re-

levancia para los ingresos.

- Spearman Correlation: que muestra la correlación con los ingresos de determinadas variables y muestra también la significación estadística basada en una prueba t. Así, la etiqueta de un jugador tiene la correlación significativa más alta con los ingresos.

El código del ejemplo se encuentra en GitHub con el nombre: “06_Ejemplo_Logit_Ordenado” en GitHub.

1.4. 4.3 Aplicación de modelos clasificación mediante clustering.

1.4.1. Usos de Aprendizaje No Supervisado

Esta sección está basada en Chang (2020) [Cha20]. A diferencia del aprendizaje supervisado, en el aprendizaje no supervisado, no hay respuestas correctas. El objetivo del aprendizaje no supervisado es identificar o simplificar la estructura de los datos. El aprendizaje no supervisado es de creciente importancia en varios campos; ejemplos se ven al agrupar a pacientes de cáncer de mama por sus marcadores genéticos, compradores por su historial de navegación y compras, y espectadores de películas por las calificaciones asignadas por los mismos espectadores.

Con este tipo de herramientas podemos querer organizar documentos en diferentes categorías mutuamente excluyentes o superpuestas, o simplemente podríamos querer visualizar los datos. A menudo es más fácil obtener datos sin etiquetar que datos etiquetados, que a menudo requieren intervención humana.

Los problemas de aprendizaje no supervisado pueden dividirse aún más en agrupación, asociación y detección de anomalías.

- Un problema de agrupación ocurre cuando queremos descubrir los agrupamientos inherentes en los datos, como agrupar a los clientes por comportamiento de compra.
- Un problema de aprendizaje de reglas de asociación (o, simplemente, asociación) es aquel en el que queremos descubrir reglas que describan conexiones en grandes porciones de nuestros datos, por ejemplo, las

personas que compran el producto A también pueden tender a comprar el producto B.

- El tercer tipo de problema, detección de anomalías o detección de valores atípicos, implica identificar elementos, eventos u observaciones que no se ajustan a un patrón esperado, como casos de fraude bancario, defectos estructurales, problemas médicos o errores en un texto. Las anomalías también se conocen como valores atípicos, novedades, ruido, desviaciones y excepciones.

Asociación

En muchas situaciones, encontrar relaciones de causalidad es el objetivo. Cuando hay un número mayor de variables, la tarea no es trivial. Sin embargo, la asociación es una condición necesaria para una relación causal. Encontrar un conjunto de eventos que correlacionen muchos otros a menudo es el punto focal y el trampolín para investigaciones futuras. El análisis de asociación proporciona una manera de encontrar el conjunto de eventos con alta densidad. Encontrar artículos de venta que están altamente relacionados (o comprados frecuentemente juntos) puede ser muy útil para el abastecimiento de estanterías, el marketing cruzado en promociones de ventas, el diseño de catálogos y la segmentación de consumidores basada en patrones de compra.

En teoría de redes, el análisis de asociación es una técnica de análisis de datos utilizada para evaluar relaciones (conexiones) entre nodos. Las relaciones pueden identificarse entre varios tipos de nodos (objetos), incluyendo organizaciones, personas y transacciones. El análisis ha sido utilizado en la investigación de actividades criminales (detección de fraude, contraterorismo e inteligencia), análisis de seguridad informática, optimización de motores de búsqueda, investigación de mercado, investigación médica e incluso en la comprensión de obras de arte.

Por ejemplo, Apriori, propuesto por Agrawal y Srikant (1994), es un algoritmo para el conjunto de ítems frecuentes sobre bases de datos transaccionales. El algoritmo procede identificando los ítems individuales frecuentes en la base de datos y extendiéndolos a conjuntos de ítems más grandes y más grandes mientras esos conjuntos de ítems aparezcan suficientemente a menudo en la base de datos. Apriori utiliza un enfoque de abajo hacia arriba, donde los subconjuntos frecuentes se extienden un ítem a la vez, y los grupos de candidatos se prueban contra los datos. El algoritmo termina cuando no

se encuentran más extensiones exitosas.¹

1.4.2. Análisis de componentes principales

El análisis de componentes principales (PCA, por sus siglas en inglés) es una herramienta importante y útil de aprendizaje no supervisado para la reducción de dimensiones. El PCA se puede utilizar como una herramienta para el preprocesamiento de datos antes de que se apliquen técnicas de aprendizaje supervisado. El PCA produce una representación de baja dimensión de un conjunto de datos. Encuentra secuencialmente un conjunto de combinaciones lineales de los predictores que tienen máxima varianza y son ortogonales entre sí.

El primer componente principal de un conjunto de características denominadas como X_1, X_2, \dots, X_K es la combinación lineal de las características o predictores X :

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \dots + \phi_{K1}X_K \quad (1.51)$$

que tiene la mayor varianza, donde ϕ_{j1} satisface:

$$\sum_{j=1}^K \phi_{j1}^2 = 1 \quad (1.52)$$

Los coeficientes ϕ_{ji} se llaman cargas y se pueden encontrar utilizando una descomposición en valores singulares de la matriz X de dimensión $n \times K$. El segundo componente principal es la combinación lineal de X_1, \dots, X_K que tiene máxima varianza entre todas las combinaciones lineales que son ortogonales con Z_1 . Así, el segundo componente sería:

$$Z_2 = \phi_{12}X_1 + \phi_{22}X_2 + \dots + \phi_{K2}X_K \quad (1.53)$$

donde el coeficiente ϕ_2 es el vector de carga del segundo componente principal, satisfaciendo $\phi_1 \cdot \phi_2 = 0$, es decir, $\phi_1 \perp \phi_2$. Este proceso continúa

¹Apriori utiliza la búsqueda en amplitud (búsqueda completa nivel por nivel) y una estructura de árbol hash para contar eficientemente los conjuntos de ítems candidatos. Genera conjuntos de ítems candidatos de longitud K a partir de conjuntos de ítems de longitud $K - 1$, y luego poda los candidatos que tienen un subpatrón infrecuente. Según la propiedad de cierre hacia abajo (si un ítem no es frecuente, cualquiera de sus superconjuntos que contengan el ítem no son frecuentes), el conjunto de candidatos contiene todos los conjuntos de ítems frecuentes de longitud k .

hasta que se encuentren todos los componentes principales $K^* = \min(n - 1, K)$. Aquí, los productos internos $\phi_i \cdot \phi_j = 0$ y $Z_i \cdot Z_j = 0$ para todos $i \neq j$.

El significado de las cargas se puede interpretar de la siguiente manera:

1. El vector de carga ϕ_1 con elementos $\phi_{11}, \phi_{21}, \dots, \phi_{K1}$ define una dirección en el espacio de características a lo largo de la cual los datos varían más.
2. Las direcciones de los componentes principales $\phi_1, \phi_2, \dots, \phi_K$ son la secuencia ordenada de vectores singulares derechos de la matriz X , y las varianzas de los componentes son $1/n$ veces los cuadrados de los valores singulares.
3. El vector de carga ϕ_1 define la línea en el espacio K -dimensional que está más cerca de las n observaciones en términos de distancia euclídea cuadrada media.

Asumiendo que las variables han sido centradas para tener media cero, la varianza total observada es:

$$\sum_{j=1}^K \text{Var}(X_j) = \sum_{j=1}^K \frac{1}{n} \sum_{i=1}^n x_{ij}^2. \quad (1.54)$$

y la varianza explicada por el m -ésimo componente principal es:

$$\text{Var}(Z_m) = \frac{1}{n} \sum_{i=1}^n z_{im}^2. \quad (1.55)$$

Por lo tanto, la varianza proporcional explicada por el m -ésimo componente principal se da por la cantidad positiva entre 0 y 1.

$$\frac{\sum_{i=1}^n z_{im}^2}{\sum_{j=1}^K \sum_{i=1}^n x_{ij}^2}. \quad (1.56)$$

Una de las desventajas de usar PCA es que las características transformadas Z no son tan intuitivas como las medidas originales X ; reduciendo las posibilidades de interpretación.

1.4.3. K-Means Clustering

El agrupamiento o clustering se refiere a un conjunto muy amplio de técnicas para encontrar subgrupos, o clústeres, en un conjunto de datos. El objetivo del agrupamiento es encontrar una partición de los datos en grupos distintos de modo que las observaciones dentro de cada grupo sean bastante similares entre sí en algún sentido. Tal sentido de similitud a menudo es una consideración específica del dominio que debe realizarse con base en el conocimiento de los datos que se estudian. La similitud debe estar relacionada con algún resultado vago o posibles múltiples resultados/propósitos.

En las bibliotecas, organizamos los libros por diferentes categorías y subcategorías, mientras que tales selecciones de categorías y subcategorías se basan en las necesidades de los clientes que a menudo no están claramente definidas. En casa, organizamos nuestras cosas en ropa, zapatos, utensilios de cocina y otros para conveniencia cuando los usamos. Por lo tanto, el agrupamiento debe tener algunos propósitos que son difíciles de caracterizar usando una medida de resultado simple. Por esta razón, el aprendizaje supervisado no es aplicable al problema.

Un buen ejemplo de agrupamiento en el comercio sería el agrupamiento para la segmentación del mercado. Supongamos que tenemos acceso a grandes datos (por ejemplo, ingreso medio del hogar, ocupación, distancia desde el área urbana más cercana) de un gran número de personas que pueden o no ser nuestros clientes existentes. Nuestro objetivo es identificar subgrupos de personas que podrían ser más receptivas a una forma particular de publicidad, o agruparlos (en términos de datos) de acuerdo con la probabilidad de comprar un producto en particular.

A diferencia del PCA, que busca una representación de baja dimensión de las observaciones que explica una buena fracción de la varianza, el agrupamiento busca subgrupos homogéneos entre las observaciones.

Hay dos métodos de agrupamiento comúnmente utilizados: el agrupamiento K-medias y el agrupamiento jerárquico. En el agrupamiento K-medias, buscamos particionar las observaciones en un número preespecificado de clústeres, mientras que en el agrupamiento jerárquico no sabemos de antemano cuántos clústeres queremos. En cambio, el agrupamiento jerárquico terminará con una representación visual en forma de árbol de las observaciones, llamada dendrograma, que nos permite ver de una vez el agrupamiento obtenido para cada número posible de clústeres (Figura 1.17).

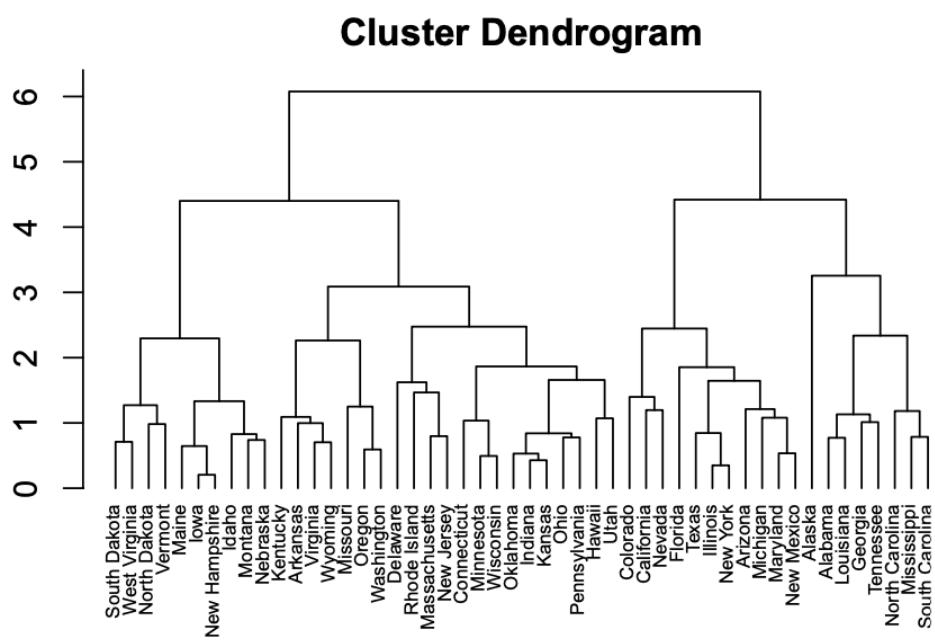


Figura 1.17: US Arrest Hierarchical Clustering, retomado de Chang (2020, p 216) [Cha20]

Algoritmo K-Means

El algoritmo K-medias es uno de los métodos de agrupamiento más simples y populares. Sea $X = \{x_i; i = 1, \dots, n\}$ y el clúster $C = \{c_j; j = 1, \dots, k\}$. Para un k dado, el objetivo del agrupamiento es encontrar C , tal que

$$\min_{c_j; j} \sum_{i=1}^n \|x_i - c_j\| \quad (1.57)$$

El algoritmo K-medias se describe de la siguiente manera:

1. Escoger aleatoriamente k puntos de datos de X como el clúster inicial C .
2. Reasignar todos $x_i \in X$ al promedio de clúster más cercano c_j .
3. Actualizar todos $c_j \in C$ con la media de sus clústeres correspondientes.
4. Repetir los pasos 2 y 3 hasta que las asignaciones de clúster no cambien.

La convergencia del algoritmo está garantizada en un número finito de iteraciones. Sin embargo, cuando la distancia $\sum_{i=1}^n \|x_i - c_j\|$ es una función no convexa, la convergencia puede llevar a un óptimo local. El algoritmo también es sensible a los valores atípicos y podría llevar a algunos clústeres vacíos. Aquí hemos asumido que k es fijo. El problema de agrupamiento con k variable es muy desafiante desde el punto de vista del cálculo.

El agrupamiento K-medias (k-medoids) ajusta exactamente k clústeres como se especifica y la asignación final de clústeres depende de los centros de clústeres iniciales elegidos y del clúster final asignado.

Clustering Jerárquico

El agrupamiento jerárquico busca construir una jerarquía de clústeres. Las estrategias para el agrupamiento jerárquico pueden ser aglomerativas o divisivas. Una estrategia aglomerativa es un enfoque de abajo hacia arriba, es decir, cada observación comienza en su propio clúster, y pares de clústeres se fusionan a medida que se asciende en la jerarquía. Una estrategia divisiva es un enfoque de arriba hacia abajo, es decir, todas las observaciones comienzan en un clúster, y se realizan divisiones de forma recursiva a medida que se desciende en la jerarquía.

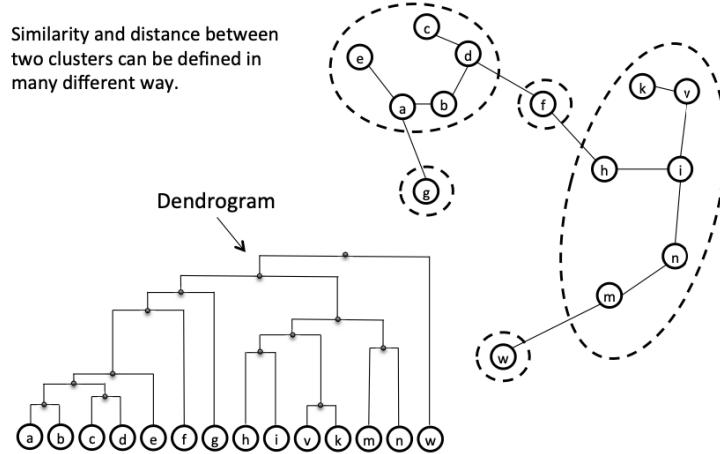


Figura 1.18: Agglomerative Clustering Algorithm and Dendrogram, retomado de Chang (2020, p 219) [Cha20]

El agrupamiento divisivo con una búsqueda exhaustiva tiene una complejidad de $O(2^n)$, pero es común utilizar heurísticas más rápidas para elegir las divisiones, como K-Medias. Los resultados del agrupamiento jerárquico generalmente se presentan en un dendrograma.

Para decidir qué clústeres se deben combinar (estrategia aglomerativa) o dónde se debe dividir un clúster (estrategia divisiva), se requiere una medida de disimilitud o similitud entre conjuntos de observaciones, como distancia o enlace entre pares de observaciones. La distancia puede ser Euclídea, Euclídea al cuadrado, Manhattan, máxima, Mahalanobis, entre otras. El criterio de enlace puede ser agrupamiento de enlace completo, $\max\{d(a, b); a \in A, b \in B\}$, agrupamiento de enlace simple, $\min\{d(a, b); a \in A, b \in B\}$, o agrupamiento de enlace promedio, $\frac{1}{|A| \cdot |B|} \sum_{a \in A} \sum_{b \in B} d(a, b)$. El enlace simple sufre de encadenamiento porque para fusionar dos grupos, solo se necesita que un par de puntos esté cerca, independientemente de todos los demás. Por lo tanto, los clústeres pueden estar demasiado dispersos. En contraste, el enlace completo evita el encadenamiento, pero sufre de hacinamiento. El enlace promedio intenta, utilizando la disimilitud promedio por pares, establecer un equilibrio.

1.4.4. 4.3.1 Método de K-means con datos de epidemiológicos.

El código del ejemplo se encuentra en GitHub con el nombre: “08_Unsupervised_Clustering_Algorithms_KMeans” en GitHub.

1.4.5. Agrupamiento de datos de series temporales

1.4.6. 4.3.2 Método de K-means para series de tiempo.

El código del ejemplo se encuentra en GitHub con el nombre: “11_Unsupervised_Clustering_Time_Series” en GitHub.

Para la recolección de datos se usan:

“09_Ejemplo_Conexion_ServicioWeb” y “10_Generando_Datos_PML” en GitHub.

1.5. 4.4 Aplicación de modelos de análisis de texto.

1.5.1. Elementos de Procesamiento de Lenguaje y Minado de Texto

En esta sección usamos principalmente el documento de Jurafsky y Martin (2023) [JM23] y el de Shmueli, et al (2019) [Shm+19].

Expresiones regulares y tokenización

Comenzaremos con la herramienta más importante para describir patrones de texto: la expresión regular. Se pueden usar expresiones regulares para especificar cadenas que podríamos querer extraer de un documento, incluso cadenas como \$199 o \$24.99 para extraer tablas de precios de un documento, por ejemplo.

Luego pasaremos a un conjunto de tareas denominadas, colectivamente, normalización de texto, en las que las expresiones regulares juegan un papel importante. Normalizar el texto significa convertirlo a un formato estándar más conveniente. Por ejemplo, la mayor parte de lo que vamos a hacer con el lenguaje se basa en separar o tokenizar primero las palabras del texto en

ejecución. Las palabras en inglés suelen estar separadas entre sí por espacios en blanco, pero los espacios en blanco no siempre son suficientes. Nueva York y rock ‘n’ roll a veces se tratan como palabras grandes a pesar de que contienen espacios, mientras que a veces tendremos que separar I’m en las dos palabras I y am. Para procesar tweets o mensajes de texto necesitaremos tokenizar emoticones como :) o hashtags como #nlproc.

Algunos idiomas, como el japonés, no tienen espacios entre palabras, por lo que la tokenización de palabras se vuelve más difícil.

Otra parte de la normalización del texto es la lematización, la tarea de determinar que dos palabras tienen la misma raíz, a pesar de sus diferencias superficiales. La lematización es esencial para procesar lenguas morfológicamente complejas como el árabe.

La derivación (stemming) se refiere a una versión más simple de lematización en la que principalmente simplemente eliminamos los sufijos del final de la palabra. La normalización del texto también incluye la segmentación de oraciones: dividir un texto en oraciones individuales, usando señales como puntos de oración o signos de exclamación.

Finalmente, necesitaremos comparar palabras y otras cadenas. Introduciremos una métrica llamada distancia de edición que mide qué tan similares son dos cadenas en función del número de ediciones (inserciones, eliminaciones, sustituciones) que se necesitan para cambiar una cadena por otra. La distancia de edición es un algoritmo con aplicaciones en todo el procesamiento del lenguaje, desde la corrección ortográfica hasta el reconocimiento de voz y la resolución de correferencia.

Expresiones regulares

Uno de los éxitos no reconocidos en la estandarización de la informática ha sido la expresión regular (a menudo abreviada como ‘regex’), un lenguaje para especificar cadenas de búsqueda de texto. Este lenguaje práctico se utiliza en todos los lenguajes informáticos, procesadores de texto y herramientas de procesamiento de textos como las herramientas Unix grep o Emacs.

Formalmente, una expresión regular es una notación algebraica para caracterizar un conjunto de cadenas. Las expresiones regulares son particularmente útiles para buscar en textos, cuando tenemos un patrón que buscar y un corpus de textos que buscar. Una función de búsqueda de expresiones regulares buscará en el corpus y devolverá todos los textos que coincidan con el patrón. El corpus puede ser un documento único o una colección. Las expresiones regulares tienen muchas variantes. Usar un probador de expresiones regulares en línea es una forma práctica de probar sus expresiones y

Regex	Example Patterns Matched
/woodchucks/	“interesting links to <u>woodchucks</u> and lemurs”
/a/	“Mary Ann stopped by Mona’s”
!/	“You’ve left the burglar behind again!” said Nori

Figure 2.1 Some simple regex searches.

Figura 1.19: Some simple regex searches, retomado de Jurafsky y Martin (2023, 3)

Regex	Match	Example Patterns
/[wW]oodchuck/	Woodchuck or woodchuck	“ <u>Woodchuck</u> ”
/[abc]/	‘a’, ‘b’, or ‘c’	“In uomini, in soldati”
/[1234567890]/	any digit	“plenty of <u>7</u> to <u>5</u> ”

Figure 2.2 The use of the brackets [] to specify a disjunction of characters.

Figura 1.20: The use of the brackets [] to specify a disjunction of characters, retomado de Jurafsky y Martin (2023, 3)

explorar estas variaciones, por ejemplo: <https://regexr.com/>.

El tipo más simple de expresión regular es una secuencia de caracteres simples; poner caracteres de concatenación en secuencia se llama concatenación. Las expresiones regulares distinguen entre mayúsculas y minúsculas. Las figuras 1.19 y 1.20.

La expresión regular /[1234567890]/ especifica cualquier dígito. Si bien clases de caracteres como dígitos o letras son componentes importantes de las expresiones, pueden resultar incómodas (por ejemplo, es inconveniente especificar /[ABCDEFGHIJKLMNPQRSTUVWXYZ]/ para que signifique “cualquier letra mayúscula”).

En los casos en los que hay una secuencia bien definida asociada con un conjunto de caracteres, los corchetes se pueden usar con el guión (-) para especificar cualquier carácter en un rango.

El patrón /[2 – 5]/ especifica cualquiera de los caracteres 2, 3, 4 o 5. El patrón /[b – g]/ especifica uno de los caracteres b, c, d, e, f o g. Algunos otros ejemplos se muestran en la figura 1.21.

Las llaves cuadradas también se pueden usar para especificar lo que no puede ser un solo carácter, mediante el uso del signo de potencia ^. Si el

Regex	Match	Example Patterns Matched
/[A-Z]/	an upper case letter	“we should call it ‘Drenched Blossoms’ ”
/[a-z]/	a lower case letter	“my beans were impatient to be hoed!”
/[0-9]/	a single digit	“Chapter 1: Down the Rabbit Hole”

Figure 2.3 The use of the brackets [] plus the dash - to specify a range.

Figura 1.21: The use of the brackets [] plus the dash - to specify a range, retomado de Jurafsky y Martin (2023, 3)

Regex	Match (single characters)	Example Patterns Matched
/[^A-Z]/	not an upper case letter	“Oyfn pipetchik”
/[^Ss]/	neither ‘S’ nor ‘s’	“I have no exquisite reason for’t”
/[^.]/	not a period	“our resident Djinn”
/[e^]/	either ‘e’ or ‘^’	“look up ^ now”
/a^b/	the pattern ‘a^b’	“look up a^b now”

Figure 2.4 The caret ^ for negation or just to mean ^. See below re: the backslash for escaping the period.

Figura 1.22: The caret ^ for negation or just to mean ^. See below re: the backslash for escaping the period, retomado de Jurafsky y Martin (2023, 3)

signo de potencia ^ es el primer símbolo después de la llave cuadrada abierta [, el patrón resultante se niega. Por ejemplo, el patrón /[^a]/ coincide con cualquier carácter (incluidos los caracteres especiales) excepto a. Esto sólo es cierto cuando el signo de intercalación es el primer símbolo después de la llave cuadrada abierta. Si aparece en cualquier otro lugar, normalmente representa un signo de potencia. La figura 1.22 muestra algunos ejemplos.

Word Tokenization

Tokenización consiste en dividir el texto en oraciones y estas, en palabras. Tokenización también se puede utilizar para expandir las contracciones que están marcadas por apóstrofes, por ejemplo, convirtiendo what’s en los dos tokens what are y we’re en we are. Algunas de estas contracciones ocurren en otros idiomas alfabéticos, incluidos artículos y pronombres en francés (j’ai, l’homme).

Dependiendo de la aplicación, los algoritmos de tokenización también pueden tokenizar expresiones de varias palabras como Nueva York o rock ’n’ roll como un solo token, lo que requiere algún tipo de diccionario de

GPT-3.5 & GPT-4 GPT-3 (Legacy)

Educación Continua FES Acatlán

Clear Show example

Tokens	Characters
10	30

Educación Continua FES Acatlán

Text Token IDs

Figura 1.23: Ejemplo de Tokenización con el modelo de OpenAI <https://platform.openai.com/tokenizer>

expresiones de varias palabras. Por lo tanto, la tokenización está íntimamente ligada al reconocimiento de entidades, la tarea de detectar nombres, fechas y organizaciones.

Un estándar de tokenización comúnmente utilizado se conoce como tokenización de Penn Treebank, utilizado para los corpus analizados (bancos de árboles) publicados por el Linguistic Data Consortium (LDC), la fuente de muchos conjuntos de datos útiles. Este estándar separa las palabras y separa toda la puntuación.

Por ejemplo, esta herramienta nos permite contar tokens y entender como separaría el texto un modelo como el de Open AI (<https://platform.openai.com/tokenizer>). La figura 1.23 muestra algunos ejemplos.

Modelos de lenguaje N-gram

Los modelos que asignan probabilidades a las siguientes palabras, o secuencias de palabras en general, se denominan modelos de lenguaje o LM. ¿Por qué queríamos predecir las próximas palabras? ¡Resulta que los grandes modelos de lenguaje que revolucionaron el PLN moderno se entranan simplemente prediciendo palabras! Los grandes modelos lingüísticos aprenden muchísimo sobre el lenguaje únicamente al entrenarlos para predecir las palabras siguientes a partir de palabras vecinas. Los modelos de lenguaje también pueden asignar una probabilidad a una oración completa.

En este capítulo presentamos el tipo de modelo de lenguaje más simple: el modelo de lenguaje de n-gramas. Un n-grama es una secuencia de n palabras: un 2-grama (que llamaremos bigrama) es una secuencia de dos palabras como “gira tu” o “tu turno”, y un 3-grama (un trígrama) es una secuencia de tres palabras como “entrega tu tarea”.

Pero también (con un poco de ambigüedad terminológica) usamos la palabra ‘n-grama’ para referirnos a un modelo probabilístico que puede estimar la probabilidad de una palabra dadas las n-1 palabras anteriores y, por lo tanto, también asignar probabilidades a secuencias completas.

La expectativa es que al final del diplomado presentaremos los modelos de redes neuronales de lenguaje grande, mucho más poderosos, basados en la arquitectura transformers. Pero debido a que los n-grams tienen una formalización notablemente simple y clara, comenzamos nuestro estudio del modelado del lenguaje con ellos, introduciendo conceptos principales que desempeñan un papel en el modelado del lenguaje, conceptos como conjuntos de prueba y entrenamiento, perplejidad, muestreo e interpolación.

N-Grams

Comencemos con la tarea de calcular $P(w|h)$, la probabilidad de una palabra w dada una historia h . Supongamos que la historia h es “su agua es tan transparente que” y queremos saber la probabilidad de que la siguiente palabra sea:

$$P(el|su\ agua\ es\ tan\ transparente\ que) \quad (1.58)$$

Una forma de estimar esta probabilidad es a partir de recuentos de frecuencia relativa. Tomemos un corpus muy grande, cuente el número de veces que vemos que “su agua es tan transparente que” y cuente el número de veces que va seguido ‘él’.

Esto sería responder a:

$$P(el|su \text{ agua es tan transparente que}) = \frac{C(\text{su agua es tan transparente que el})}{C(\text{su agua es tan transparente que})} \quad (1.59)$$

Con un corpus lo suficientemente grande, como la web, podemos calcular estos recuentos y estimar la probabilidad a partir de la ecuación (1.59). Parece una tarea titánica.

Por esta razón, necesitaremos introducir formas más inteligentes de estimar la probabilidad de una palabra w dada una historia h , o la probabilidad de una secuencia completa de palabras W . Comencemos formalizando un poco la notación.

Para representar la probabilidad de que una variable aleatoria particular X_i tome el valor “el”, o $P(X_i = el)$, usaremos la simplificación $P(el)$. Representaremos una secuencia de n palabras como $w_1 \dots w_n$ o $w_{1:n}$.

Para la probabilidad conjunta de que cada palabra en una secuencia tenga un valor particular $P(X_1 = w_1, X_2 = w_2, X_3 = w_3, \dots, X_n = w_n)$ usaremos $P(w_1, w_2, \dots, w_n)$.

Una cosa que podemos hacer es descomponer esta probabilidad usando la regla de la cadena de probabilidad:

$$\begin{aligned} P(X_1, \dots, X_n) &= P(X_1)P(X_2|X_1)P(X_3|X_{1:2})\dots P(X_n|X_{1:n-1}) \\ &= \prod_{k=1}^n P(X_k|X_{1:k-1}) \end{aligned} \quad (1.60)$$

Bayes ingenuo - Naive Bayes, clasificación de texto y sentimiento

Muchas tareas de procesamiento del lenguaje implican clasificación. En esta sección presentamos el algoritmo Naive Bayes y lo aplicamos a la categorización de texto, la tarea de asignar una etiqueta o categoría a un texto o documento completo.

Nos centramos en una tarea común de categorización de textos, el análisis de sentimientos, la extracción de sentimientos, la orientación positiva o negativa que un escritor expresa hacia algún objeto. Una reseña de una película, un libro o un producto en la web expresa el sentimiento del autor hacia el producto, mientras que un texto editorial o político expresa un sentimiento hacia un candidato o una acción política. Por lo tanto, extraer el sentimiento del consumidor o del público es relevante para campos que van desde el marketing hasta la política.

La versión más simple del análisis de sentimientos es una tarea de clasificación binaria, y las palabras de la revisión brindan excelentes pistas.

La detección de spam es otra aplicación comercial importante, la tarea de clasificación binaria de asignar un correo electrónico a una de las dos clases de spam o no spam. Se pueden utilizar muchas características léxicas y de otro tipo para realizar esta clasificación. Por ejemplo, podría sospechar razonablemente de un correo electrónico que contenga frases como “SIN NINGÚN COSTO” o “Estimado Ganador”.

El objetivo de la clasificación es tomar una sola observación, extraer algunas características útiles y así clasificar la observación en una de un conjunto de clases discretas. Un método para clasificar texto es utilizar reglas escritas a mano por humanos.

La forma más común de clasificar textos en el procesamiento del lenguaje es mediante el aprendizaje automático supervisado, el tema de este capítulo. En el aprendizaje supervisado, tenemos un conjunto de datos de observaciones de entrada, cada una asociada con alguna salida correcta (una “señal de supervisión”). El objetivo del algoritmo es aprender cómo mapear desde una nueva observación hasta una salida correcta.

Formalmente, la tarea de la clasificación supervisada es tomar una entrada x y un conjunto fijo de clases de salida $Y = \{y_1, y_2, \dots, y_M\}$ y devolver una clase predicha $y \in Y$. Para la clasificación de texto, a veces hablemos de c (para “clase”) en lugar de y como nuestra variable de salida, y d (para “documento”) en lugar de x como nuestra variable de entrada.

En la situación supervisada tenemos un conjunto de entrenamiento de N documentos, cada uno de los cuales ha sido etiquetado manualmente con una clase: $\{(d_1, c_1), \dots, (d_N, c_N)\}$. Nuestro objetivo es aprender un clasificador que sea capaz de mapear desde un nuevo documento d a su clase correcta $c \in C$, donde C es un conjunto de clases de documentos útiles. Un clasificador probabilístico además nos dirá la probabilidad de que la observación esté en la clase. Esta distribución completa entre las clases puede ser información útil para decisiones posteriores; evitar tomar decisiones discretas desde el principio puede resultar útil al combinar sistemas.

Se utilizan muchos tipos de algoritmos de aprendizaje automático para crear clasificadores. Ahora presentamos el algoritmo Naive Bayes; el siguiente discutimos la regresión logística. Estos ejemplifican dos formas de hacer la clasificación. Los clasificadores generativos como el Naive Bayes construyen un modelo de cómo una clase podría generar algunos datos de entrada. Dada una observación, devuelven la clase que tiene más probabilidades de

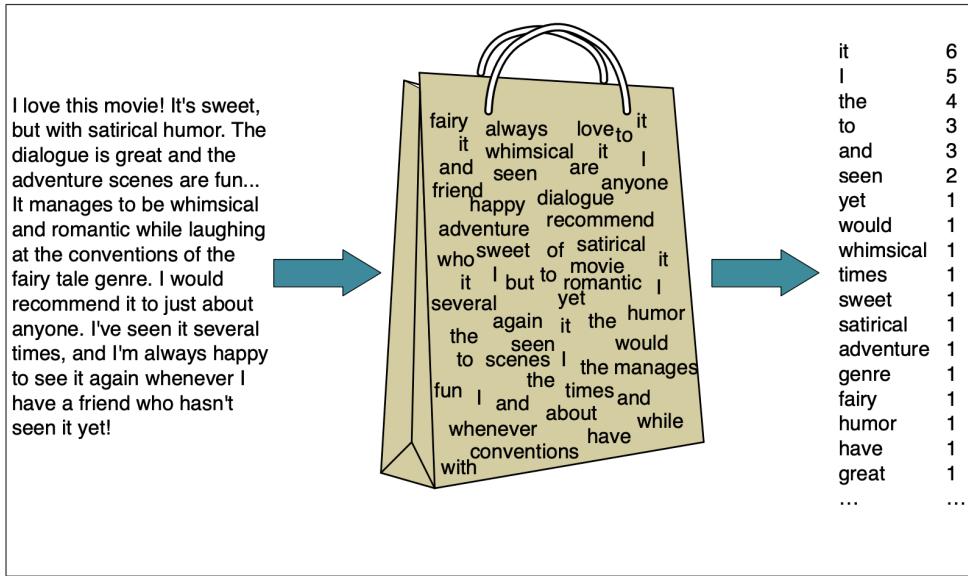


Figure 4.1 Intuition of the multinomial naive Bayes classifier applied to a movie review. The position of the words is ignored (the *bag-of-words* assumption) and we make use of the frequency of each word.

Figura 1.24: Retomado de Jurafsky y Martin (2023)

haber generado la observación. En cambio, los clasificadores discriminativos como la regresión logística aprenden qué características de la entrada son más útiles para discriminar entre las diferentes clases posibles. Si bien los sistemas discriminativos suelen ser más precisos y, por tanto, más utilizados, los clasificadores generativos siguen desempeñando un papel.

Naive Bayes Classifiers

Usaremos el clasificador multinomial Naive Bayes. La intuición del clasificador se muestra en la figura 1.24, donde representamos un documento de texto como si fuera una bolsa de palabras, es decir, un conjunto desordenado de palabras ignorando su posición, manteniendo solo su frecuencia en el documento. En el ejemplo de la figura, en lugar de representar el orden de las palabras en todas las frases como “Me encanta esta película” y “La recomendaría”, simplemente observamos que la palabra “I” apareció 5 veces en la extracto completo, la palabra “it” 6 veces, las palabras y así sucesivamente.

Naive Bayes es un clasificador probabilístico, lo que significa que para un documento d , de todas las clases $c \in C$, el clasificador devuelve la clase \hat{c} que tiene la probabilidad posterior máxima dado el documento. En la siguiente

ecuación utilizamos la notación hat ^ para referirnos a “nuestra estimación de la clase correcta”.

$$\hat{c} = \arg \max_{c \in C} P(c|d) \quad (1.61)$$

La intuición de la clasificación bayesiana es utilizar la regla de Bayes para transformar la ecuación (1.61) en otras probabilidades que tienen algunas propiedades útiles. La regla de Bayes se presenta en la ecuación (1.62); nos da una manera de descomponer cualquier probabilidad condicional $P(x|y)$ en otras tres probabilidades:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} \quad (1.62)$$

Podemos sustituir la ecuación (1.62) en la ecuación (1.61):

$$\hat{c} = \arg \max_{c \in C} P(c|d) = \arg \max_{c \in C} \frac{P(d|c)P(c)}{P(d)} \quad (1.63)$$

Podemos simplificar convenientemente la ecuación (1.63) eliminando el denominador $P(d)$. Esto es posible porque calcularemos $P(d|c)P(c)/P(d)$ para cada clase posible. Pero $P(d)$ no cambia para cada clase. Siempre estamos preguntando por la clase más probable para el mismo documento d , que debe tener la misma probabilidad $P(d)$. Así, podemos elegir la clase que maximice esta fórmula más simple:

$$\hat{c} = \arg \max_{c \in C} P(c|d) = \max_{c \in C} P(d|c)P(c) \quad (1.64)$$

Llamamos a Naive Bayes como un modelo generativo porque podemos leer la ecuación (1.64) establece una especie de suposición implícita sobre cómo se genera un documento: primero se toma una muestra de una clase a partir de $P(c)$, y luego las palabras se generan mediante una muestra de $P(d|c)$. (De hecho, podríamos imaginarnos generar documentos artificiales, o al menos su recuento de palabras, siguiendo este proceso).

Para volver a la clasificación, calculamos la clase más probable \hat{c} dado algún documento d eligiendo la clase que tiene el producto más alto de dos probabilidades. La probabilidad previa de la clase $P(c)$ y la probabilidad del documento $P(d|c)$:

$$\hat{c} = \arg \max_{c \in C} \overbrace{P(d|c)}^{likelihood} \overbrace{P(c)}^{prior} \quad (1.65)$$

Sin pérdida de generalidad, podemos representar un documento d como un conjunto de características f_1, f_2, \dots, f_n :

$$\hat{c} = \arg \max_{c \in C} \overbrace{P(f_1, f_2, \dots, f_n | c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}} \quad (1.66)$$

Desafortunadamente, la ecuación (1.66) todavía es demasiado difícil de calcular directamente. Sin algunas suposiciones simplificadoras, estimar la probabilidad de cada combinación posible de características (por ejemplo, cada conjunto posible de palabras y posiciones) requeriría una gran cantidad de parámetros y conjuntos de entrenamiento increíblemente grandes. Por lo tanto, los clasificadores Naive Bayes parten de dos supuestos simplificadores.

La primera es la suposición de la bolsa de palabras discutida intuitivamente anteriormente. Asumimos que la posición no importa y que la palabra “amor” tiene el mismo efecto en la clasificación ya sea que aparezca como la primera, vigésima o última palabra en el documento. Por lo tanto, asumimos que las características f_1, f_2, \dots, f_n solo codifican la identidad de la palabra y no la posición.

La segunda se denomina comúnmente supuesto ingenuo de Bayes. Se trata del supuesto de independencia condicional de que las probabilidades $P(f_i | c)$ son independientes dada la clase c y, por tanto, pueden multiplicarse “ingenuamente” de la siguiente manera:

$$P(f_1, f_2, \dots, f_n | c) = P(f_1 | c) \cdot P(f_2 | c) \cdot \dots \cdot P(f_n | c) \quad (1.67)$$

Así, la ecuación final para la clase seleccionada por el clasificador Naive Bayes será:

$$c_{NB} = \arg \max_{c \in C} P(c) \prod_{f \in F} P(f | c) \quad (1.68)$$

Para aplicar el clasificador Naive Bayes al texto, debemos considerar las posiciones de las palabras, simplemente recorriendo un índice a través de cada posición de las palabras en el documento:

posiciones \leftarrow todas las posiciones de palabras en el documento de prueba

$$c_{NB} = \arg \max_{c \in C} P(c) \prod_{i \in \text{Posiciones}} P(w_i | c) \quad (1.69)$$

Los cálculos del clasificador Naive Bayes, al igual que los cálculos para el modelado del lenguaje, se realizan en el *log* espacio para evitar el desbordamiento y aumentar la velocidad. Así, la ecuación (1.69) generalmente se

expresa como:

$$c_{NB} = \arg \max_{c \in C} \log P(c) + \sum_{i \in \text{Posiciones}} \log P(w_i|c) \quad (1.70)$$

Al considerar características en el espacio logarítmico, la ecuación (1.70) calcula la clase predicha como una función lineal de las características de entrada. Los clasificadores que utilizan una combinación lineal de entradas para tomar una decisión de clasificación (como el Naive Bayes y también la regresión logística) se denominan clasificadores lineales.

Training the Naive Bayes Classifier

¿Cómo podemos conocer las probabilidades $P(c)$ y $P(f_i|c)$? Consideremos primero la estimación de máxima verosimilitud. Simplemente usaremos las frecuencias en los datos. Para la clase prior $P(c)$ preguntamos qué porcentaje de los documentos en nuestro conjunto de entrenamiento están en cada clase c . Sea N_c el número de documentos en nuestros datos de entrenamiento con clase c y N_{doc} el número total de documentos. Entonces:

$$\hat{P}(c) = \frac{N_c}{N_{doc}} \quad (1.71)$$

Para conocer la probabilidad $P(f_i|c)$, asumiremos que una característica es simplemente la existencia de una palabra en la bolsa de palabras del documento, por lo que querremos $P(w_i|c)$, que calculamos como la fracción de veces aparece la palabra w_i entre todas las palabras en todos los documentos de la clase c . Primero concatenamos todos los documentos con categoría c en un gran texto de “categoría c”. Luego usamos la frecuencia de w_i en este documento concatenado para dar una estimación de máxima verosimilitud de la probabilidad:

$$\hat{P}(w_i|c) = \frac{\text{count}(w_i, c)}{\sum_{w \in V} \text{count}(w, c)} \quad (1.72)$$

Aquí el vocabulario V consiste en la unión de todos los tipos de palabras de todas las clases, no sólo de las palabras de una clase c .

Sin embargo, existe un problema con el entrenamiento de máxima verosimilitud. Imaginemos que estamos tratando de estimar la probabilidad de que la palabra “fantástico” dado la clase positivo, pero supongamos que no hay documentos de entrenamiento que contengan la palabra “fantástico” y estén clasificados como positivo. Quizás la palabra “fantástico” aparezca

(¿sarcásticamente?) en la clase negativo. En tal caso, la probabilidad de esta característica será cero:

$$\hat{P}(\text{"fantastico"}|\text{positivo}) = \frac{\text{count}(\text{"fantastico"}, \text{positivo})}{\sum_{w \in V} \text{count}(w, \text{positivo})} = 0 \quad (1.73)$$

Regresión lógistica

Este tipo de técnicas se suelen ilustrar con esta analogía. Los cuentos de detectives están tan llenos de pistas como los textos lo están de palabras. Sin embargo, para el pobre lector puede ser un desafío saber cómo ponderar las pistas del autor para realizar la crucial tarea de clasificación: decidir quién es el culpable.

La regresión logística es una de las herramientas analíticas más importantes en las ciencias sociales y naturales. En el procesamiento del lenguaje natural, la regresión logística es el algoritmo de aprendizaje automático supervisado de línea base para la clasificación, y también tiene una relación muy estrecha con las redes neuronales—una red neuronal puede ser vista como una serie de clasificadores de regresión logística apilados uno encima del otro. Así, las técnicas de clasificación y aprendizaje automático introducidas aquí jugarán un papel importante e su formación de científicos de datos.

La regresión logística se puede utilizar para clasificar una observación en una de dos clases (como ‘sentimiento positivo’ y ‘sentimiento negativo’), o en una de varias clases. Debido a que los marcos para las tareas de clasificación más simples son más simples, describiremos este caso especial de regresión logística en las siguientes secciones, y luego resumiremos brevemente el uso de la regresión logística multinomial para más de dos clases.

Clasificadores Generativos y Discriminativos. La diferencia más importante entre naive Bayes y regresión logística es que la regresión logística es un **clasificador discriminativo** mientras que naive Bayes es un **clasificador generativo**.

Estos son dos marcos diferentes para cómo construir un modelo de aprendizaje automático. Considera una metáfora visual: imagina que estamos tratando de distinguir imágenes de perros de imágenes de gatos. Un modelo generativo tendría que comprender cómo se ven los perros y los gatos. Literalmente podrías pedirle a dicho modelo que ‘genere’, es decir, dibuje, un perro. Dado una imagen de prueba, el sistema luego pregunta si es el modelo de gato o el modelo de perro el que mejor se ajusta (está menos sorprendido por) la imagen, y elige eso como su etiqueta.

Un modelo discriminativo, por otro lado, solo está tratando de aprender a distinguir las clases (quizás sin aprender mucho sobre ellas). Entonces, tal vez todos los perros en el conjunto de datos de entrenamiento llevan collares y los gatos no. Si esa única característica separa claramente las clases, entonces el modelo está satisfecho. Si le preguntas a un modelo así lo que sabe sobre los gatos, todo lo que puede decir es que no llevan collares.

Recordemos que el modelo Bayes ingenuo asigna una clase c a un documento d no por el cálculo directo de $P(c|d)$, sino calculando una verosimilitud y un prior:

$$\hat{c} = \arg \max_{c \in C} \overbrace{P(d|c)}^{likelihood} \overbrace{P(c)}^{prior} \quad (1.74)$$

Un modelo generativo como el Naive Bayes utiliza este término de verosimilitud, que expresa cómo generar las características de un documento *si supiéramos que es de la clase c* . Por el contrario, un modelo discriminativo en este escenario de categorización de texto intenta calcular directamente $P(c|d)$. Quizás aprenda a asignar un gran peso a las características del documento que mejoran directamente su habilidad para discriminar entre clases posibles, incluso si no pudiera generar un ejemplo de una de las clases.

Componentes de un clasificador de aprendizaje automático probabilístico. Como el Naive Bayes, la regresión logística es un clasificador probabilístico dentro del aprendizaje supervisado. Los clasificadores de aprendizaje automático requieren un corpus de entrenamiento de m pares entrada/salida $(x^{(i)}, y^{(i)})$. (Utilizaremos superíndices en paréntesis para referirnos a instancias individuales en el conjunto de entrenamiento –para la clasificación de sentimientos cada instancia podría ser un documento individual a ser clasificado). Un sistema de aprendizaje automático para clasificación tiene entonces cuatro componentes:

1. Una **representación de características** de entrada. Para cada observación de entrada $x^{(i)}$, esto será un vector de características $[x_1, x_2, \dots, x_n]$. Nos referiremos generalmente a la característica i para la entrada $x^{(i)}$ como $x_j^{(i)}$, a veces simplificado como x_i , pero también veremos la notación f_i , $f(x)$, o, para clasificación multiclas, $f_i(c, x)$.
2. Una **función de clasificación** que computa, para la clase estimada, vía $p(y|x)$. Más delante introduciremos las herramientas de la función sigmoide y softmax para clasificación.

3. Una **función objetivo** para el aprendizaje, usualmente involucrando minimizar el error en ejemplos de entrenamiento. Introduciremos la función de pérdida de entropía cruzada.
4. Un **algoritmo para optimizar la función objetivo**. Introducimos el algoritmo de descenso de gradiente estocástico.

La regresión logística tiene dos fases:

Entrenamiento: Entrenamos el sistema (específicamente los pesos w y el sesgo b) utilizando descenso de gradiente estocástico y la pérdida de entropía cruzada.

Prueba: Dado un ejemplo de prueba x , calculamos $p(y|x)$ y devolvemos la probabilidad más alta ya sea $y = 1$ o $y = 0$.

La función sigmoide

El objetivo de la regresión logística binaria es entrenar un clasificador que pueda tomar una decisión binaria sobre la clase de una nueva observación de entrada. Aquí introducimos la función **sigmoide** que ayudará a tomar esta decisión.

Considera una sola observación de entrada x , la cual representaremos por un vector de características $[x_1, x_2, \dots, x_n]$. El clasificador de salida será 1 (significando que la observación es miembro de la clase) o 0 (la observación no es miembro de la clase). Queremos saber la probabilidad $P(y = 1|x)$ de que la observación sea miembro de la clase. Así que la decisión es “positivo” versus “negativo”, las características representan conteos de palabras en un documento, $P(y = 1|x)$ es la probabilidad de que el documento tenga sentimiento positivo, y $P(y = 0|x)$ es la probabilidad de que el documento tenga sentimiento negativo.

La regresión logística resuelve esta tarea aprendiendo, de un conjunto de entrenamiento, un vector de pesos y un término de sesgo. Cada peso w_i está asociado con una de las características de entrada x_i . El peso w_i representa cuán importante es esa característica de entrada para la decisión de clasificación, y puede ser positivo (proporcionando evidencia de que la instancia que se clasifica pertenece a la clase positiva) o negativo (proporcionando evidencia de que la instancia que se clasifica pertenece a la clase negativa). Así, esperaríamos que en una tarea de análisis de sentimiento la palabra **increíble** tenga un peso positivo alto, y **pésimo** un peso negativo muy alto. El término

de sesgo, también llamado el **intercepto**, es otro número real que se añade a las entradas ponderadas.

Para tomar una decisión en una instancia de prueba –después de haber aprendido los pesos en el entrenamiento– el clasificador primero multiplica cada x_i por su peso w_i , suma las características ponderadas, y añade el término de sesgo b . El número único resultante z expresa la suma ponderada de la evidencia para la clase.

$$z = \left(\sum_{i=1}^n w_i x_i \right) + b \quad (1.75)$$

En adelante representaremos tales sumas usando la notación de **producto punto** del álgebra lineal. El producto punto de dos vectores \mathbf{a} y \mathbf{b} , escrito como $\mathbf{a} \cdot \mathbf{b}$, es la suma de los productos de los elementos correspondientes de cada vector. Por lo tanto, lo siguiente es una representación equivalente a la Ecuación (1.75):

$$z = \mathbf{w} \cdot \mathbf{x} + b \quad (1.76)$$

Pero nótese que nada en la Ecuación (1.76) obliga a z a ser una probabilidad, es decir, a estar entre 0 y 1. De hecho, dado que los pesos son valores reales, la salida podría ser incluso negativa; z varía de $-\infty$ a ∞ .

Para crear una probabilidad, pasaremos z a través de la función **sigmoide**, $\sigma(z)$. La función sigmoide (llamada así porque se parece a una 's') también se llama la función **logística**, y le da su nombre a la regresión logística. La sigmoide tiene la siguiente ecuación, y se muestra gráficamente en la Figura 1.25:

$$\sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + \exp(-z)} \quad (1.77)$$

La función sigmoide tiene varias ventajas; toma un número real y lo mapea dentro del rango $(0,1)$, que es justo lo que queremos para una probabilidad. Porque es casi lineal alrededor del 0 pero se aplana hacia los extremos, tiende a aplastar valores atípicos hacia 0 o 1. Es diferenciable.

Si aplicamos el sigmoide a la suma de las características ponderadas, obtenemos un número entre 0 y 1. Para hacerlo una probabilidad, solo necesitamos asegurarnos de que los dos casos, $p(y = 1)$ y $p(y = 0)$, sumen 1.

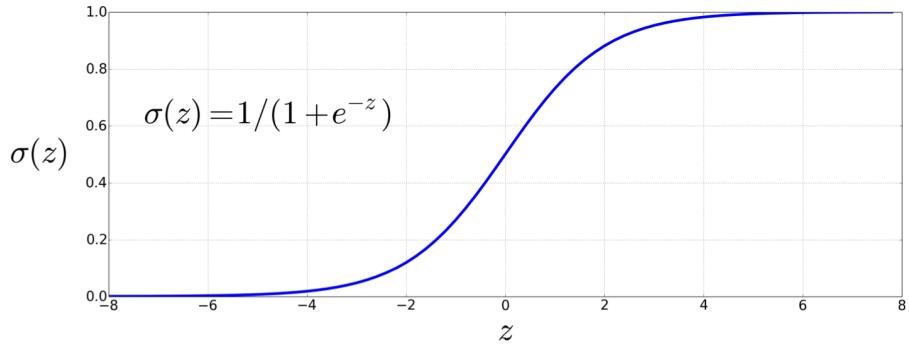


Figura 1.25: La función sigmoide $\sigma(z) = \frac{1}{1+e^{-z}}$ toma un valor real y lo mapea al rango $(0, 1)$. Es casi lineal alrededor de 0, pero los valores atípicos se comprimen hacia 0 o 1. Retomado de Jurafsky y Martin (2023)

Podemos hacer esto de la siguiente manera:

$$\begin{aligned}
 P(y = 1) &= \sigma(\mathbf{w} \cdot \mathbf{x} + b) \\
 &= \frac{1}{1 + \exp(-(\mathbf{w} \cdot \mathbf{x} + b))} \\
 P(y = 0) &= 1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b) \\
 &= 1 - \frac{1}{1 + \exp(-(\mathbf{w} \cdot \mathbf{x} + b))} \\
 &= \frac{\exp(-(\mathbf{w} \cdot \mathbf{x} + b))}{1 + \exp(-(\mathbf{w} \cdot \mathbf{x} + b))} \tag{1.78}
 \end{aligned}$$

La función sigmoide tiene la propiedad de que

$$1 - \sigma(x) = \sigma(-x) \tag{1.79}$$

así que también podríamos haber expresado $P(y = 0)$ como $\sigma(-(\mathbf{w} \cdot \mathbf{x} + b))$.

La entrada a la función sigmoide, el puntaje $z = \mathbf{w} \cdot \mathbf{x} + b$ (de la Ecación (1.76)), a menudo se llama el **logit**. Esto es porque la función logit es la inversa de la sigmoide. La función logit es el logaritmo de la razón de probabilidades $\frac{p}{1-p}$:

$$\text{logit}(p) = \sigma^{-1}(p) = \ln \left(\frac{p}{1-p} \right) \tag{1.80}$$

Usar el término **logit** para z es una forma de recordarnos que al usar el sigmoide para convertir z (que varía de $-\infty$ a ∞) en una probabilidad, estamos interpretando implícitamente a z no solo como cualquier número real, sino específicamente como un logaritmo de probabilidades.

Clasificación con Regresión Logística.

La función sigmoide de la sección anterior nos proporciona una manera de tomar una característica x y calcular la probabilidad $P(y = 1|x)$. ¿Cómo tomamos una decisión sobre qué clase aplicar a una característica de prueba x ? Para un x dado, decimos que sí si la probabilidad $P(y = 1|x)$ es mayor a 0.5, y no en caso contrario. Llamamos a esto el **límite de decisión**:

$$\text{decisión}(x) = \begin{cases} 1 & \text{si } P(y = 1|x) > 0.5 \\ 0 & \text{en caso contrario} \end{cases} \quad (1.81)$$

Veamos algunos ejemplos de aplicación de la regresión logística como clasificador para tareas de lenguaje.

Clasificación de Sentimientos

Supongamos que estamos haciendo clasificación de sentimientos binarios en textos de reseñas de películas, y nos gustaría saber si asignar la clase de sentimiento positivo o negativo a un documento de prueba.

Var	Definition	Value in Fig. 1.26
x_1	count(positive lexicon words \in doc)	3
x_2	count(negative lexicon words \in doc)	2
x_3	$\begin{cases} 1 & \text{if "no" } \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns \in doc)	3
x_5	$\begin{cases} 1 & \text{if "?" } \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	ln(word count of doc)	ln(66) = 4.19

Supongamos por el momento que ya hemos aprendido un peso real para cada característica de la función logística que hemos descrito.

Cada una de estas características, y los 6 pesos correspondientes a las 6 características son $[2.5, -5.0, -1.2, 0.5, 2.0, 0.7]$, mientras que $b = 0.1$. (Más adelante discutiremos cómo se aprenden los pesos). El peso w_1 , por ejemplo, indica qué tan importante es la característica del número de palabras positivas del léxico (como *genial*, *agradable*, *disfrutable*, etc.) para una decisión de sentimiento positivo, mientras que w_2 nos dice la importancia de las

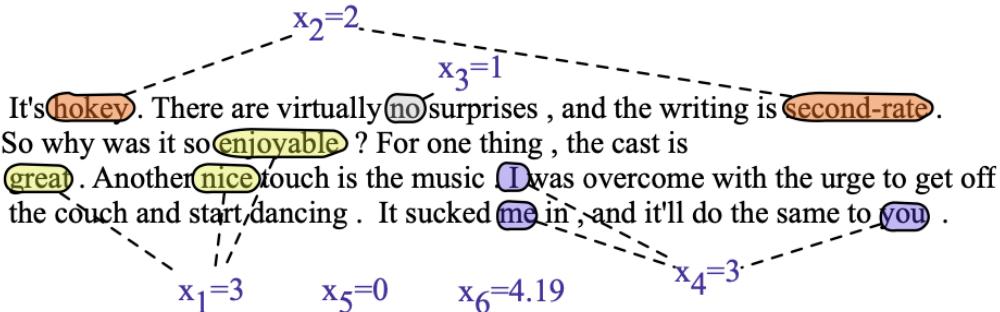


Figura 1.26: Un documento de prueba mini muestra que exhibe las características extraídas en el vector x . Retomado de Jurafsky y Martin (2023)

palabras negativas del léxico. Nota que $w_1 = 2.5$ es positivo, mientras que $w_2 = -5.0$, lo que significa que las palabras negativas están negativamente asociadas con una decisión de sentimiento positivo, y son aproximadamente el doble de importantes que las palabras positivas.

Dadas estas 6 características y la revisión de entrada x , $P(+|x)$ y $P(-|x)$ se pueden calcular usando la Ecuación 1.78:

$$\begin{aligned}
 p(+|x) &= P(y = 1|x) = \sigma(\mathbf{w} \cdot \mathbf{x} + b) \\
 &= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\
 &= \sigma(.833) \\
 &= 0.70
 \end{aligned} \tag{1.82}$$

$$\begin{aligned}
 p(-|x) &= P(y = 0|x) = 1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b) \\
 &= 0.30
 \end{aligned}$$

Otras tareas de clasificación y características

La regresión logística se aplica comúnmente a todo tipo de tareas de PLN, y cualquier propiedad de la entrada puede ser una característica. Considera la tarea de desambiguación de puntos: decidir si un punto es el final de una oración o parte de una palabra, clasificando cada punto en una de dos clases EOS (fin-de-oración) y no-EOS. Podríamos usar características como x_i abajo expresando que la palabra actual está en minúsculas (quizás con un peso

positivo), o que la palabra actual está en nuestro diccionario de abreviaturas (“Prof.”) (quizás con un peso negativo). Una característica también puede expresar una combinación bastante compleja de propiedades. Por ejemplo, un punto después de una palabra en mayúscula es probablemente un EOS, pero si la palabra misma es *St.* y la palabra anterior está en mayúsculas, entonces el punto probablemente es parte de una abreviación de la palabra *street*.

$$x_1 = \begin{cases} 1 & \text{si } \text{Case}(w_i) = \text{Lower} \\ 0 & \text{en caso contrario} \end{cases} \quad (1.83)$$

$$x_2 = \begin{cases} 1 & \text{si } w_i \in \text{AcronymDict} \\ 0 & \text{de lo contrario} \end{cases} \quad (1.84)$$

$$x_3 = \begin{cases} 1 & \text{si } w_i = \text{St.} \text{ y } \text{Case}(w_{i-1}) = \text{Cap} \\ 0 & \text{de lo contrario} \end{cases} \quad (1.85)$$

Diseñando características: Las características generalmente se diseñan examinando el conjunto de entrenamiento con un ojo en las intuiciones lingüísticas y la bibliografía existente en el dominio. Un análisis cuidadoso de errores en el conjunto de entrenamiento o en una versión temprana de un sistema a menudo proporciona información sobre las características.

Para algunas tareas es especialmente útil construir características complejas que son combinaciones de características más primitivas. Vimos tal característica para la desambiguación de períodos anteriormente, donde un punto en la palabra *St.* era menos probable que fuera el final de la oración si la palabra anterior estaba capitalizada. Las características de combinación o *interacciones de características* tienen que ser diseñadas a mano.

Para muchas tareas (especialmente cuando los valores de las características pueden referenciar palabras específicas) se necesitan grandes números de características. A menudo se crean automáticamente a partir de plantillas de características, especificaciones abstractas de características. Por ejemplo, una plantilla de bigrama podría especificar que se cree una característica para cada par de palabras antes de un punto en el conjunto de entrenamiento. Por lo tanto, el espacio de características es disperso, ya que solo creamos una característica si ese n-grama existe en esa posición en el conjunto de entrenamiento. La característica generalmente se crea como un hash del string en las descripciones. Un descriptor de la característica como “bigram(American

“breakfast)” se convierte en un número único entero i que se convierte en el número de característica f_i .

Para evitar el extensivo esfuerzo humano en el diseño de características, la investigación reciente en PLN se ha centrado en el *aprendizaje de representación*: formas de aprender características automáticamente de una manera no supervisada a partir de la entrada—este tema escapa de este diplomado, pero es una buena opción para estudios posteriores—.

Estandarizando las características de entrada: Cuando diferentes características de entrada tienen rangos de valores extremadamente diferentes, es común reescalarlos para que tengan rangos comparables. Estandarizamos los valores de entrada centrando los valores para que resulten en una media cero y una desviación estándar de uno (esta transformación a veces se llama la transformación-z). Es decir, si μ_i es la media de los valores de la característica x_i a través de las m observaciones en el conjunto de datos de entrada, y σ_i es la desviación estándar

$$\begin{aligned}\mu_i &= \frac{1}{m} \sum_{j=1}^m x_i^{(j)} \\ \sigma_i &= \sqrt{\frac{1}{m} \sum_{j=1}^m (x_i^{(j)} - \mu_i)^2} \\ x'_i &= \frac{x_i - \mu_i}{\sigma_i}\end{aligned}$$

Alternativamente, podemos **normalizar** los valores de las características de entrada para que estén entre 0 y 1:

$$x'_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$$

Tener datos de entrada con un rango comparable es útil al comparar valores entre características. **El escalado de datos es especialmente importante en grandes redes neuronales, ya que ayuda a acelerar el descenso de gradiente.**

Procesando muchos ejemplos a la vez

Hemos mostrado las ecuaciones para la regresión logística para un solo ejemplo. Pero en la práctica, por supuesto querremos procesar un conjunto de pruebas entero con muchos ejemplos. Supongamos que tenemos un conjunto

de pruebas consistente en m ejemplos de prueba cada uno de los cuales nos gustaría clasificar. Continuaremos usando la notación inicial, en la cual un superíndice entre paréntesis se refiere al índice del ejemplo en algún conjunto de datos (ya sea para entrenamiento o para pruebas). Así que en este caso cada ejemplo de prueba $x^{(i)}$ tiene un vector de características $x^{(i)}$, $1 \leq i \leq m$. (Como es usual, representaremos vectores y matrices en negrita).

Una forma de calcular cada valor de salida $y^{(i)}$ es simplemente tener un bucle ‘for’ y calcular cada ejemplo de prueba uno a la vez:

$$\begin{aligned} & \textbf{foreach } x^{(i)} \text{ in input } [x^{(1)}, x^{(2)}, \dots, x^{(m)}] \\ & \quad y^{(i)} = \sigma(\mathbf{w} \cdot x^{(i)} + b) \end{aligned} \tag{1.86}$$

Para los primeros 3 ejemplos de prueba, entonces, estaríamos calculando las predicciones $y^{(i)}$ de la siguiente manera:

$$\begin{aligned} P(y^{(1)} = 1 | x^{(1)}) &= \sigma(\mathbf{w} \cdot x^{(1)} + b) \\ P(y^{(2)} = 1 | x^{(2)}) &= \sigma(\mathbf{w} \cdot x^{(2)} + b) \\ P(y^{(3)} = 1 | x^{(3)}) &= \sigma(\mathbf{w} \cdot x^{(3)} + b) \end{aligned}$$

Pero resulta que podemos modificar ligeramente nuestra ecuación original Ecuación (1.76) para hacer esto mucho más eficientemente. Usaremos aritmética de matrices para asignar una clase a todos los ejemplos con una sola operación de matriz.

Primero, empaquetaremos todos los vectores de características de entrada para cada entrada x en una única matriz de entrada X , donde cada fila i es un vector fila consistiendo en el vector de características para el ejemplo de entrada $x^{(i)}$ (es decir, el vector $x^{(i)}$). Asumiendo que cada ejemplo tiene f características y pesos, X será entonces una matriz de forma $[m \times f]$, de la siguiente manera:

$$X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_f^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_f^{(2)} \\ x_1^{(3)} & x_2^{(3)} & \cdots & x_f^{(3)} \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix} \tag{1.87}$$

Ahora, si introducimos b como un vector de longitud m que consiste en el sesgo escalar término \mathbf{b} repetido m veces, $b = [\mathbf{b}, b, \dots, b]$, y \mathbf{y} como el vector

de salidas (un escalar $y^{(i)}$ para cada entrada $x^{(i)}$ y su vector de características $x^{(i)}$), y representamos el peso \mathbf{w} como un vector columna, podemos calcular todas las salidas con una sola multiplicación de matriz y una adición.

Ahora, si introducimos \mathbf{b} como un vector de longitud m que consiste en el término de sesgo escalar b repetido m veces, $\mathbf{b} = [b, b, \dots, b]$, y $\hat{\mathbf{y}} = [\hat{y}^{(1)}, \hat{y}^{(2)}, \dots, \hat{y}^{(m)}]$ como el vector de salidas (un escalar $\hat{y}^{(i)}$ para cada entrada $x^{(i)}$ y su vector de características $x^{(i)}$), y representamos el vector de pesos w como un vector columna, podemos computar todas las salidas con una sola multiplicación de matrices y una adición:

$$y = \mathbf{X}\mathbf{w} + \mathbf{b} \quad (1.88)$$

Esta ecuación calcula lo mismo que nuestro bucle for en la Ecuación (1.86). Por ejemplo, $\hat{y}^{(1)}$, la primera entrada del vector de salida y , será correctamente:

$$\hat{y}^{(1)} = \left[x_1^{(1)}, x_2^{(1)}, \dots, x_f^{(1)} \right] \cdot \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_f \end{bmatrix} + \mathbf{b} \quad (1.89)$$

Así, la Ecuación 1.87 con las dimensiones mostradas:

$$\begin{aligned} y &= \mathbf{X}\mathbf{w} + \mathbf{b} \\ (m \times 1) &= (m \times f) (f \times 1) + (m \times 1) \end{aligned} \quad (1.90)$$

1.5.2. 4.4.1 Ejemplos y aplicaciones de minado de texto.

El código del ejemplo se encuentra en GitHub con el nombre: “12_Expresiones_Regulares” y “13_Leer_Datos_PDF” en GitHub.

1.5.3. 4.4.2 Implementación de modelo de lenguaje natural con N-Gramas.

El código del ejemplo se encuentra en GitHub con el nombre: “14_N_Grams” en GitHub.

1.5.4. 4.4.3 Clasificación de texto con Naive Bayes.

El código del ejemplo se encuentra en GitHub con el nombre:
“15_Naive_Bayes” en GitHub.

1.5.5. 4.4.3 Clasificación de texto con regresión lógistica.

El código del ejemplo se encuentra en GitHub con el nombre:
“16_Logistic_Regression” en GitHub.

2

Temas Selectos para el Análisis de Datos

2.1. 6.1 Modelos de árboles de decisión y bosques aleatorios.

Métodos basados en árboles

En esta sección discutiremos los métodos basados en árboles para regresión y clasificación. Estos métodos implican estratificar o segmentar el espacio de los predictores en un número J de regiones. Podemos ver el ejemplo de la Figura 2.1.

En esta sección introduciremos las técnicas de bagging, random forest y boosting. Cada uno de ellos involucra múltiples árboles, los cuales son combinados para alcanzar un consenso en la predicción.

Bases de los árboles de decisión. Estos pueden aplicar a casos de regresión y clasificación. Una forma simple de verlo es como se muestra en la Figura 2.1 muestra el proceso de segmentación a través de un árbol.

En estos ejemplos, esas R_1, R_2 y R_3 son conocidas como nodos terminales u hojas del árbol. El proceso de delimitación de esos nodos terminales se lleva a cabo en 2 etapas:

1. Dividimos los predictores (X_1, X_2, \dots, X_p) en J regiones distinguibles, que denominamos R_1, R_2, \dots, R_J .
2. Para cada observación i que se ubique dentro de cada región R_j hacemos

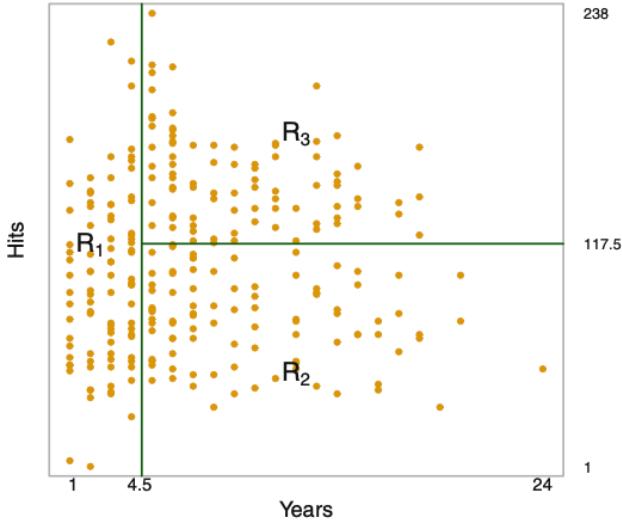


FIGURE 8.2. The three-region partition for the **Hitters** data set from the regression tree illustrated in Figure 8.1.

Figura 2.1: Ejemplo de regiones. Retomado de Hastie et al (2017, 305)

la misma predicción, la cual puede ser simplemente la media de los valores de la variable explicada en el conjunto de entrenamiento.

Pero ¿cómo construimos las regiones? La respuesta es responder a encontrar el valor mínimo o minimizar la suma de errores al cuadrado (RSS, por sus siglas en inglés) dadas las R_1, R_2, \dots, R_J :

$$RSS = \sum_{j=1}^J \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 \quad (2.1)$$

Donde \hat{y}_{R_j} es la media de las observaciones o variable explicada de entrenamiento dentro de cada región j -ésima. La implementación se puede ver como el siguiente proceso de 'división binaria recursiva':

- Seleccionamos el predictor X_j que se parte en las regiones: $\{X|X_j < s\}$ y $\{X|X_j \geq s\}$
- Corremos este análisis para todos los predictores X_1, X_2, \dots, X_p y para cada cuota s para cada predictor y región R_j .

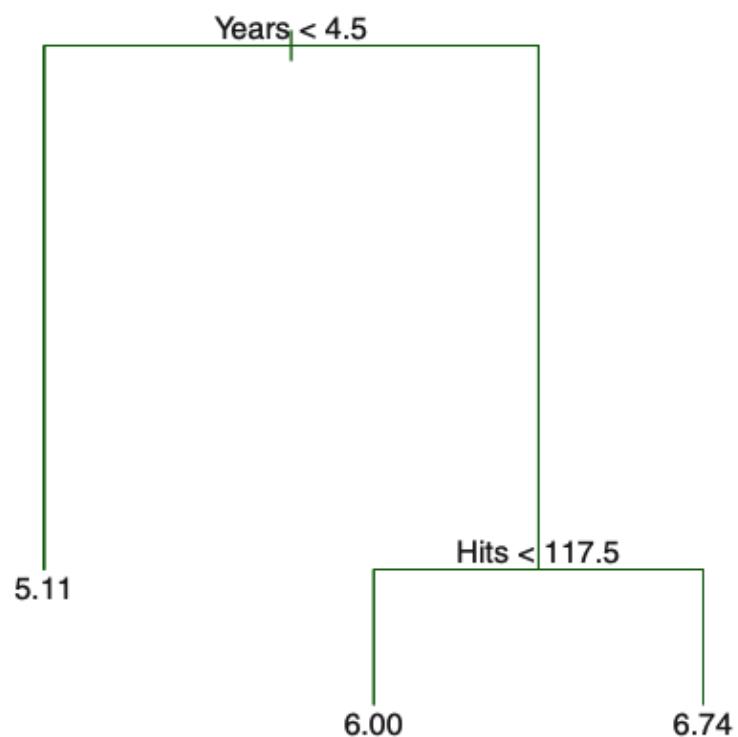


Figura 2.2: Ejemplo de árbol. Retomado de Hastie et al (2017, 304)

Ejemplo de Random Forest

El código del ejemplo se encuentra en GitHub con el nombre: “01_Decision_Trees_and_Random_Forests_Example” en GitHub.

2.2. 6.2 Modelos de redes neuronales.

2.2.1. Introducción y Motivación

El procedimiento de estimación mediante redes neuronales es un método que puede ser usado tanto para clasificación como para predicción. En ocasiones, este es considerado como una caja negra en términos de la interpretación de los parámetros estimados.

Quizá la principal ventaja de este método es que tiene una amplia exactitud o capacidad predictiva. Por esta razón se suele considerar que las redes neuronales son una herramienta computacional fundamental.

Las redes neuronales (también denominadas redes neuronales artificiales) están basadas en el funcionamiento biológico del cerebro, donde las neuronas están interconectadas y aprenden de la experiencia.

El origen de la técnica se ubica en el trabajo de McCulloch - Pitts (1943), en el cual se propone un modelo simplificado de una neurona basada en el funcionamiento biológico de las neuronas con el objetivo de establecer un resultado en términos de una proposición lógica.

Su estructura soporta/captura relaciones altamente complejas entre las variables usadas como predictores y la variable objetivo o explicada. Situación que no siempre es posible con otros métodos.

El uso moderno de las redes neuronales es llamado aprendizaje profundo (deep learning), ya que las redes modernas son profundas (deep), es decir, que tienen múltiples capas ocultas.

2.2.2. Algunos conceptos relevantes

En comparación con un proceso de regresión, las redes neuronales no requieren que los usuarios/analistas determinen una forma funcional o forma específica de la relación entre los predictores y la variable objetivo. En cambio, las redes neuronales tratan de aproximar/inferir/determinar la relación a partir de los datos.

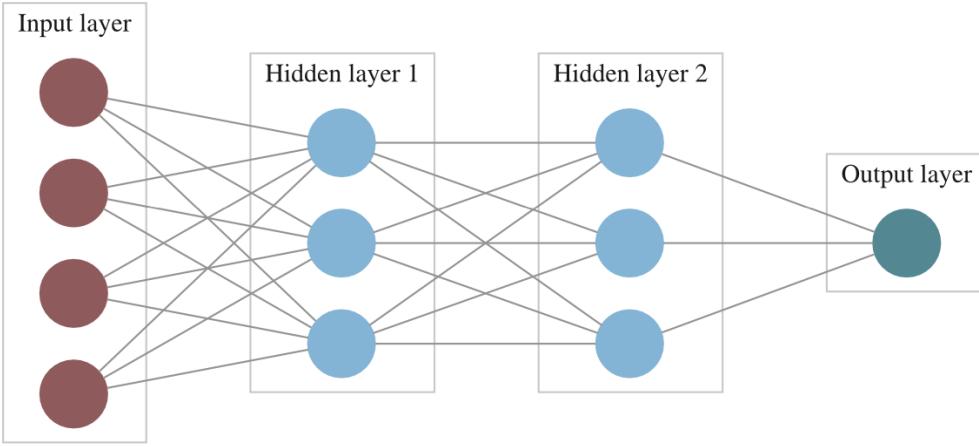


Figura 2.3: Ejemplo de redes neuronales. Retomado de Shmueli et al (2019, 285) [Shm+19]

De hecho, la regresión lineal y logística son un caso particular de las redes neuronales en la que solo hay una capa que conjunta a los inputs y los outputs, y no hay capas ocultas. Las capas ocultas son aquellas que se ubican entre la capa de input y la capa de output. La Figura 2.3 ilustra esta estructura de capas.

Otros conceptos relevantes son los nodos input y output. Si tenemos p predictores, la capa de input usualmente incluirá p -nodos. Así, las capas ocultas tienen como su input al output de la capa input, y así sucesivamente.

Unidades neuronales - neurona. En su corazón, una unidad neuronal (neurona) toma la suma ponderada de sus inputs con un término adicional (término de sesgo), θ_j .

Dado un conjunto de inputs $\{x_1, x_2, \dots, x_p\}$, una unidad toma un conjunto de pesos $\theta_j, w_{1j}, w_{2j}, \dots, w_{pj}$. Para computar el output de una capa de nodos oculta calculamos la suma ponderada de los input y aplicamos una función de activación a ellos. Es decir, dado un conjunto de valores input $\{x_1, x_2, \dots, x_p\}$ calculamos el output del nodo $j = 1, 2, \dots, J$:

$$Z_j = \theta_j + \sum_{i=1}^p w_{ij}x_i = \theta_j + \mathbf{W}_j \cdot \mathbf{X} \quad (2.2)$$

Donde los pesos $\theta_j, w_{1j}, w_{2j}, \dots, w_{pj}$ se obtienen mediante el aprendizaje.

Adicionalmente, θ_j también es conocido como el sesgo del nodo j , es una constante que controla el nivel de la contribución del nodo j .

Finalmente, aplicamos una función que denominaremos como función de activación para una unidad a . Así, podemos implementar una función de activación para el nodo j del cual obtenemos el output:

$$Output_j = a = f(Z_j) \quad (2.3)$$

Función que puede ser, por ejemplo, una función logística:

$$Output_j = f(\theta_j + \mathbf{W}_j \cdot \mathbf{X}) = \frac{1}{e^{-(\theta_j + \sum_{i=1}^p w_{ij}x_i)}} = \frac{1}{e^{-(\theta_j + \mathbf{W}_j \cdot \mathbf{X})}} \quad (2.4)$$

En este punto debemos observar que el proceso de determinación de los pesos $\theta_j, w_{1j}, w_{2j}, \dots, w_{pj}$ es iterativo y suele considerar como valores iniciales a valores pequeños aleatorios. Para simplificar asumiremos que el término de sesgo, θ_j , está contenido como uno de los predictores \mathbf{X} y se determina como parte de los pesos \mathbf{W}_j , de esta forma:

$$Z_j = \sum_{i=0}^p w_{ij}x_i = \mathbf{W}_j \cdot \mathbf{X} \quad (2.5)$$

Donde, el predictor ubicado en la posición $i = 0$ será un uno 1.

En relación a la función de activación, en la práctica se suele hacer uso de 3:

1. Sigmoide, la cual se ilustra en la Figura 2.4.

$$y = \sigma(Z) = \frac{1}{1 + e^{-Z}} \quad (2.6)$$

2. tanh, la cual se ilustra en la Figura 2.5.

$$y = \tanh(Z) = \frac{e^Z - e^{-Z}}{e^Z + e^{-Z}} \quad (2.7)$$

3. ReLU (Rectified Linear Unit), la cual se ilustra en la Figura 2.5.

$$y = ReLU(Z) = \max Z, 0 \quad (2.8)$$

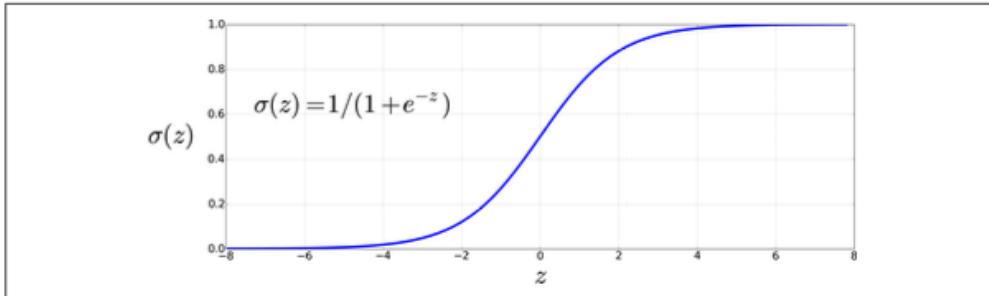


Figure 7.1 The sigmoid function takes a real value and maps it to the range $(0, 1)$. It is nearly linear around 0 but outlier values get squashed toward 0 or 1.

Figura 2.4: Ejemplo de la función de activación Sigmoide. Retomado de Jurafsky, et al (2024, Cap. 7, p.2) [JM23]

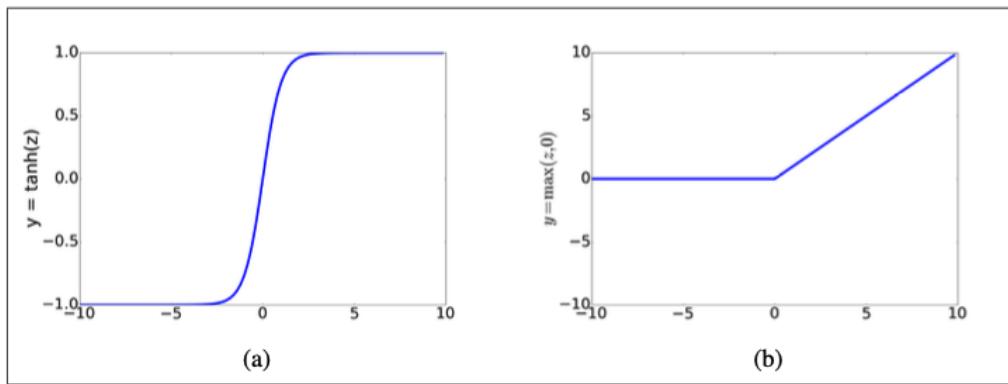


Figure 7.3 The tanh and ReLU activation functions.

Figura 2.5: Ejemplo de la función de activación *tanh*. Retomado de Jurafsky, et al (2024, Cap. 7, p.4) [JM23]

Tomemos el siguiente ejemplo. Supongamos los valores para un nodo j :

$$\mathbf{W}_j = [0.5 \ 0.2 \ 0.3 \ 0.9]$$

$$\mathbf{X} = \begin{bmatrix} 1 \\ 0.5 \\ 0.6 \\ 0.1 \end{bmatrix}$$

$$\begin{aligned} y &= \sigma(\mathbf{W}_j \cdot \mathbf{X}) \\ &= \frac{1}{1 + e^{-(\mathbf{W}_j \cdot \mathbf{X})}} \\ &= \frac{1}{1 + e^{-(0.5 + 0.2 \times 0.5 + 0.3 \times 0.6 + 0.9 \times 0.1)}} \\ &= \frac{1}{1 + e^{-(0.87)}} \\ &= 0.7 \end{aligned}$$

En términos simples, este proceso aplica para cada uno de los casos en las Figuras 2.6 y 2.7.

Una unidad neuronal muy simple es conocida como el perceptrón, el cual tiene como salida una respuesta binaria:

$$y = \begin{cases} 0 & \text{si } \mathbf{W}_j \cdot \mathbf{X} \leq 0 \\ 1 & \text{si } \mathbf{W}_j \cdot \mathbf{X} > 0 \end{cases} \quad (2.9)$$

2.2.3. Preprocesamiento de datos

Cuando usamos una función de activación logística las redes neuronales se desempeñan mejor cuando los predictores y la variable objetivo están en el intervalo $[0, 1]$; considerando:

$$X_{\text{normalizada}} = \frac{X - a}{b - a} \quad (2.10)$$

Donde $a = \min(X)$ y $b = \max(X)$.

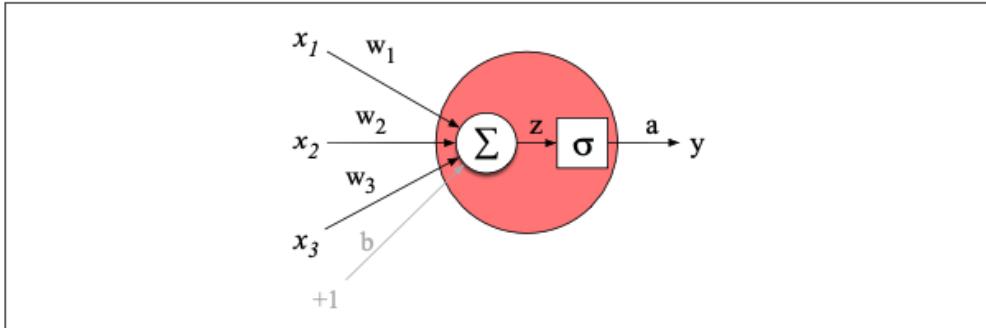


Figure 7.2 A neural unit, taking 3 inputs x_1 , x_2 , and x_3 (and a bias b that we represent as a weight for an input clamped at $+1$) and producing an output y . We include some convenient intermediate variables: the output of the summation, z , and the output of the sigmoid, a . In this case the output of the unit y is the same as a , but in deeper networks we'll reserve y to mean the final output of the entire network, leaving a as the activation of an individual node.

Figura 2.6: Ilustración de una unidad neuronal. Retomado de Jurafsky, et al (2024, Cap. 7, p.3) [JM23]

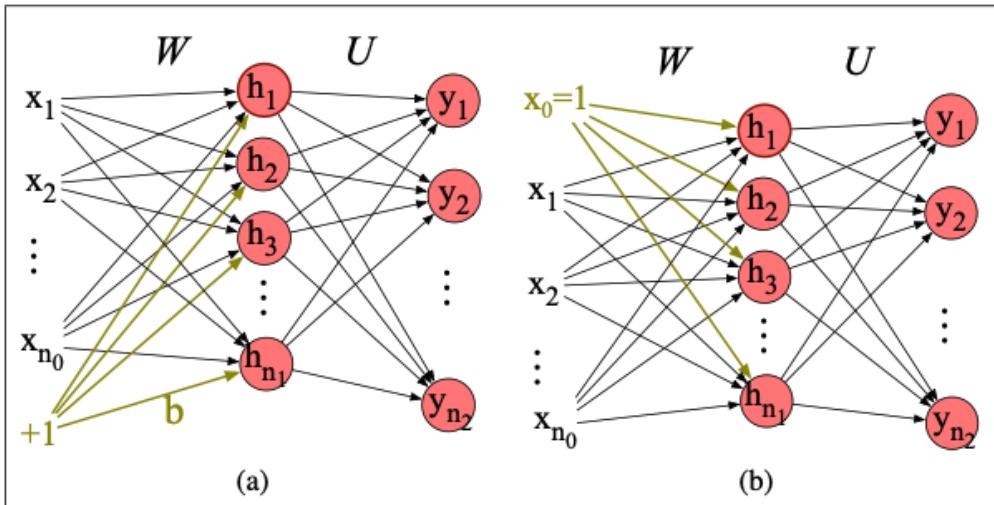


Figure 7.9 Replacing the bias node (shown in a) with x_0 (b).

Figura 2.7: Ilustración de una red neuronal con 1 capa oculta. Retomado de Jurafsky, et al (2024, Cap. 7, p.1) [JM23]

Adicionalmente, existe una función de normalización de los valores contenidos en los vectores que se forman en cada nodo Z_1, Z_2, \dots, Z_d . Esta función es conocida como función Softmax:

$$\text{Softmax}(Z_j) = \frac{z_j}{\sum_{i=1}^d e^{z_i}} \quad (2.11)$$

Donde $1 \leq i \leq d$.

2.2.4. Entrenamiento del modelo

Entrenar un modelo significa estimar los pesos $\theta_j, w_{1j}, w_{2j}, \dots, w_{pj}$ que llevan al mejor resultado de predicción posible.

El mecanismo seguido para este proceso se le conoce como **back propagation**. Es el método más popular usado para determinar los pesos (aprender). Como su nombre lo indica, los errores son calculados en forma recursiva; iniciando en la última capa (la capa de output) y continuando con la previa hasta llegar a la capa de input.

El proceso de entrenamiento se puede pensar como los siguientes pasos:

1. Establecemos una función de pérdida que modele la distancia entre el output conseguido y el el objetivo. A esa función se le suele llamar **función de entropía cruzada**.
2. Para encontrar los parámetros que minimicen la función de pérdida, usaremos el algoritmo de optimización del gradiente descendente.
3. Finalmente, implementamos un algoritmo de **backpropagation**.

2.2.5. Ejemplo de Redes Neuronales

El código del ejemplo se encuentra en GitHub con el nombre: “02_Introduccion_a_Deep_Learning” en GitHub.

2.3. 6.3 Aplicaciones de modelos de lenguaje natural con bases vectoriales (word embeddings).

Los Word Embeddings o Vectores de Embeddings son una representación vectorial del texto. Esta representación es usada en los modelos de lenguaje

natural basados en redes neuronales. Los Vectores de Embeddings pueden capturar el significado y contexto de palabras y frases.

Word embeddings:

1. Word embeddings o simplemente embeddings son el resultado de un proceso mediante el cual se utiliza alguna técnica/algoritmo que sea capaz de convertir palabras o texto a vectores de N dimensiones.
2. Estos vectores contienen cierto nivel de información semántica sobre el texto o palabra.
3. Por ejemplo, palabras que son muy similares van a tener valores cercanos en sus representaciones en vectores.
4. Existen diversos modelos que son capaces de hacer un embedding de un texto, en este caso utilizaremos el embedding de OpenAI, el cual tiene la capacidad de posicionar muy bien palabras o textos según su semántica.
5. Este es el mismo embedding que utiliza GPT3 y GPT4.

Visistar: <https://projector.tensorflow.org/>

2.4. 6.4 Introducción a los LMMs y aplicaciones basadas en LLMs (Prompt Engineering).

2.4.1. Guideline for Prompt engineering

Introducción (o ¿Qué es el procesamiento del lenguaje natural?)

En los últimos modelos de IA están desbloqueando áreas en las que el texto es abundante. En este momento, el procesamiento del lenguaje natural (PNL o NPL) es una de las áreas más populares de la inteligencia artificial (IA) gracias a aplicaciones como generadores de texto que:

- componen ensayos coherentes
- chatbots que engañan a las personas haciéndoles creer que son sensibles

Creating Embeddings

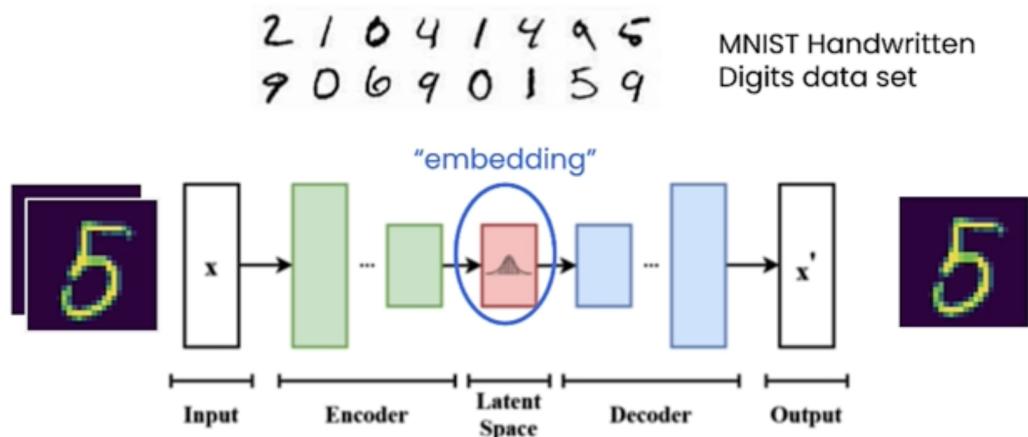


Figura 2.8: Retomado de <https://learndeeplearning.ai/courses/vector-databases-embeddings-applications/lesson/1/introduction>

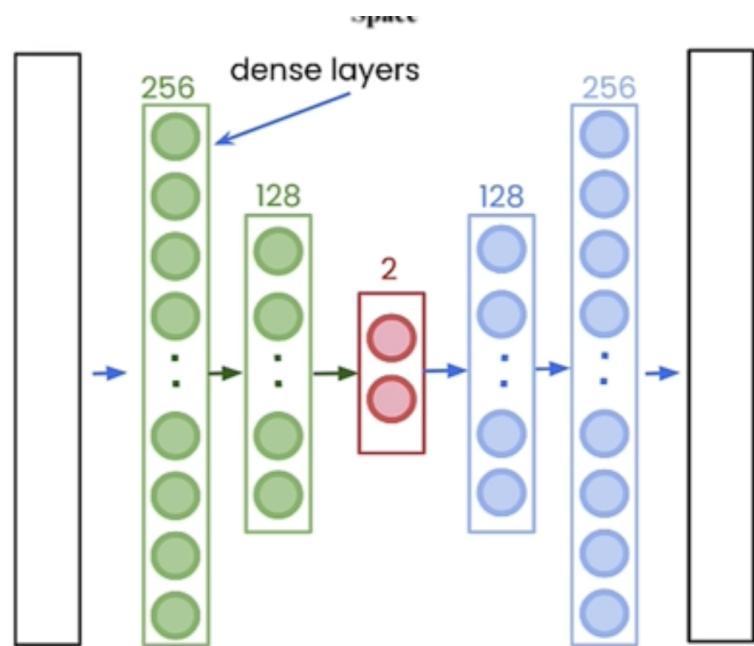


Figura 2.9: Retomado de <https://learndeeplearning.ai/courses/vector-databases-embeddings-applications/lesson/1/introduction>

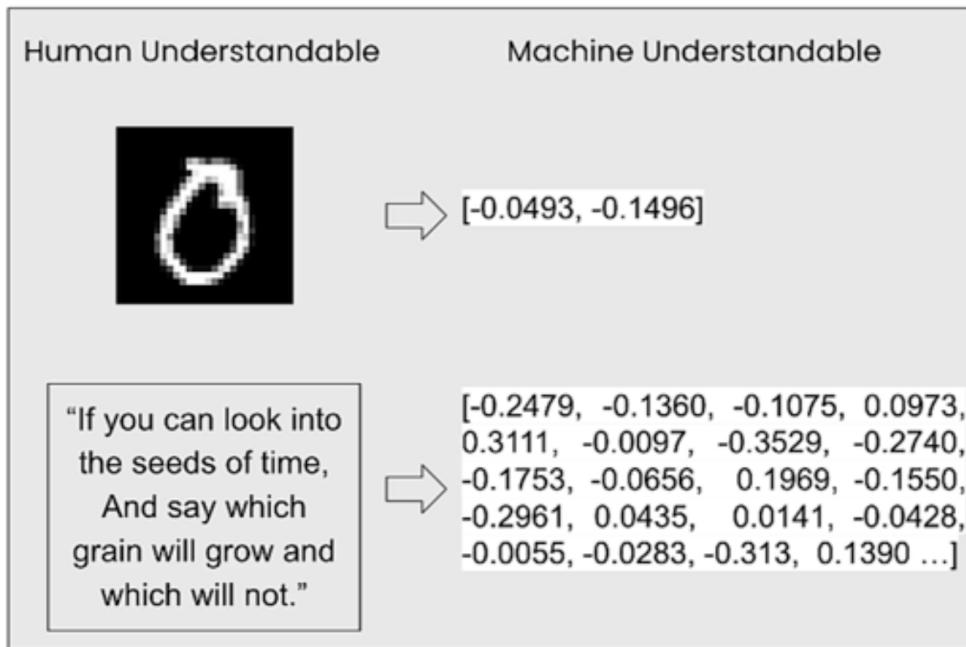


Figura 2.10: Retomado de <https://learn.deeplearning.ai/courses/vector-databases-embeddings-applications/lesson/1/introduction>



Figure 6.1 A two-dimensional (t-SNE) projection of embeddings for some words and phrases, showing that words with similar meanings are nearby in space. The original 60-dimensional embeddings were trained for sentiment analysis. Simplified from Li et al. (2015) with colors added for explanation.

Figura 2.11: Retomado de Jurafsky, et al (2024, Cap. 6, p.6) [JM23]

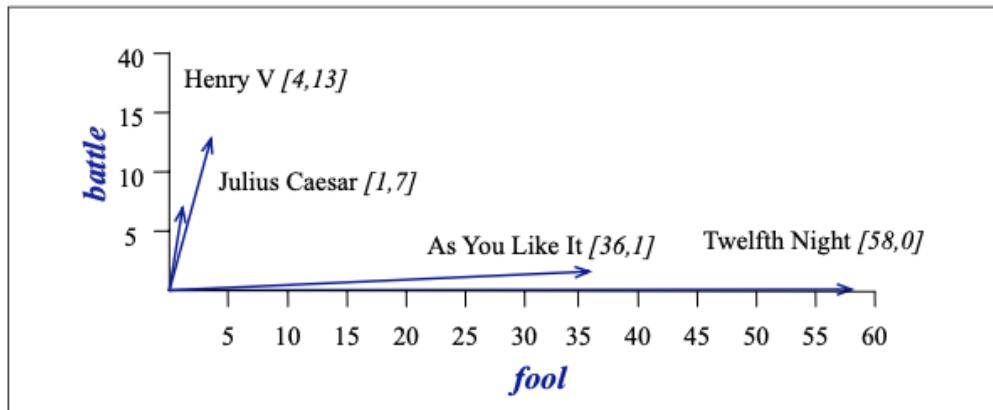


Figure 6.4 A spatial visualization of the document vectors for the four Shakespeare play documents, showing just two of the dimensions, corresponding to the words *battle* and *fool*. The comedies have high values for the *fool* dimension and low values for the *battle* dimension.

Figura 2.12: Retomado de Jurafsky, et al (2024, Cap. 6, p.8) [JM23]

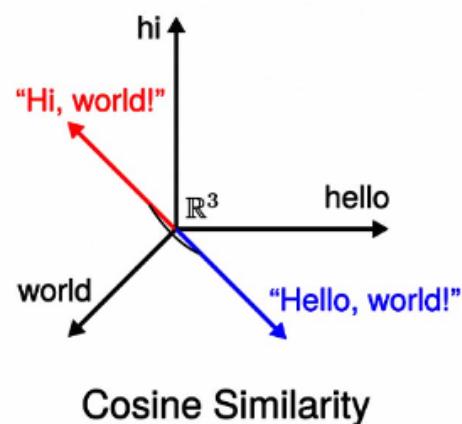


Figura 2.13: Similaridad de Coseno

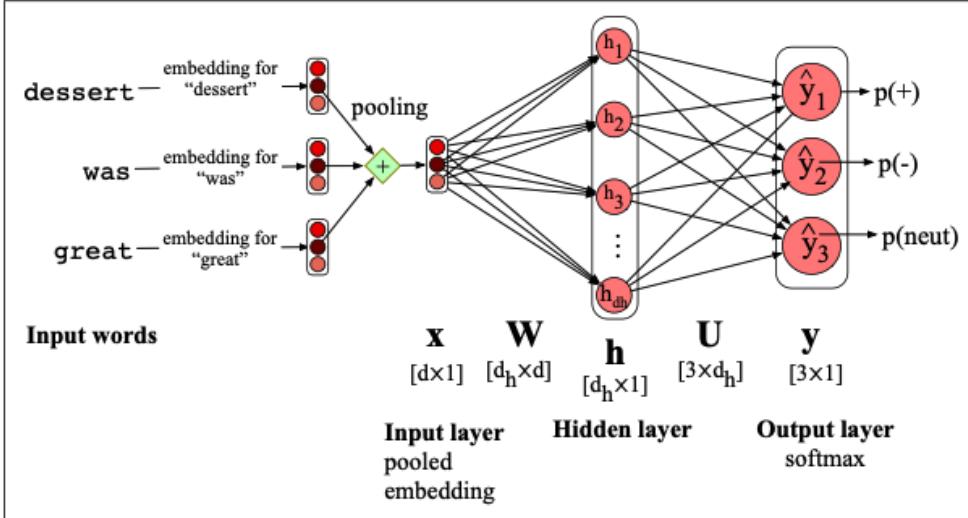


Figure 7.11 Feedforward network sentiment analysis using a pooled embedding of the input words.

Figura 2.14: Retomado de Jurafsky, et al (2024, Cap. 7, p.13) [JM23]

- programas de conversión de texto a imagen que producen imágenes fotorrealistas

¿Qué es el procesamiento del lenguaje natural?

El procesamiento del lenguaje natural es la disciplina que busca construir máquinas que puedan manipular el lenguaje humano (o datos que se asemejan al lenguaje humano) en la forma en que se escribe, se habla y se organiza.

Evolucionó a partir de la lingüística computacional, que utiliza la informática para comprender los principios del lenguaje, pero en lugar de desarrollar marcos teóricos, el NLP es una disciplina de ingeniería que busca desarrollar tecnología para realizar tareas útiles.

El NLP se puede dividir en dos subcampos superpuestos:

- Comprensión del lenguaje natural, que se centra en el análisis semántico o la determinación del significado previsto del texto, y
- Generación del lenguaje natural, que se centra en la generación de texto por una máquina.

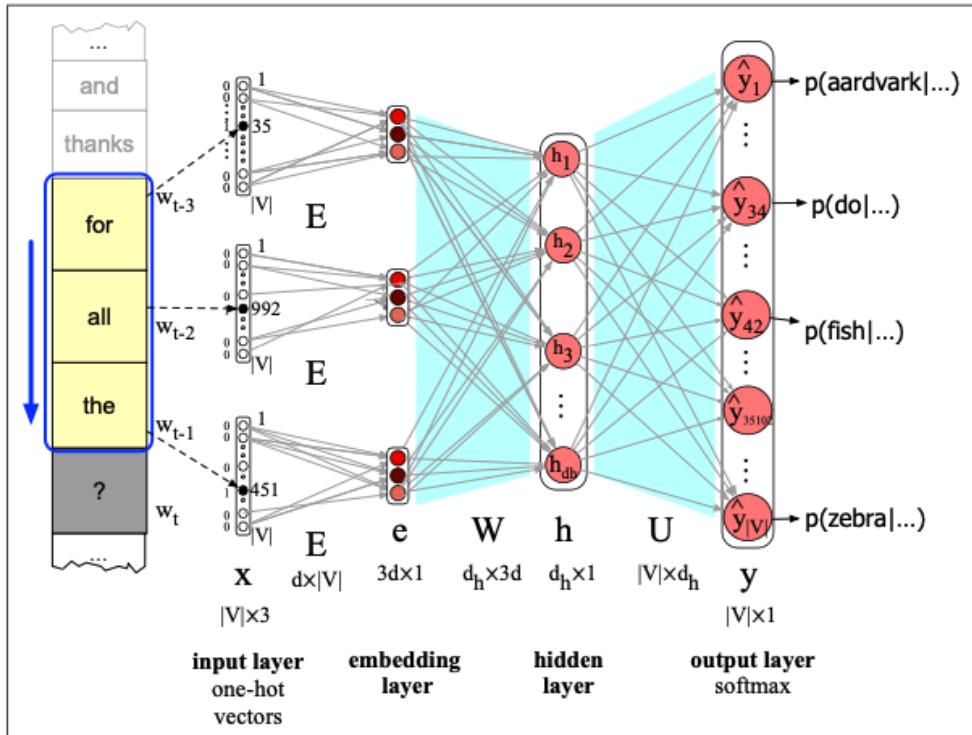


Figure 7.17 Forward inference in a feedforward neural language model. At each timestep t the network computes a d -dimensional embedding for each context word (by multiplying a one-hot vector by the embedding matrix \mathbf{E}), and concatenates the 3 resulting embeddings to get the embedding layer \mathbf{e} . The embedding vector \mathbf{e} is multiplied by a weight matrix \mathbf{W} and then an activation function is applied element-wise to produce the hidden layer \mathbf{h} , which is then multiplied by another weight matrix \mathbf{U} . Finally, a softmax output layer predicts at each node i the probability that the next word w_t will be vocabulary word V_i .

Figura 2.15: Retomado de Jurafsky, et al (2024, Cap. 7, p.23) [JM23]

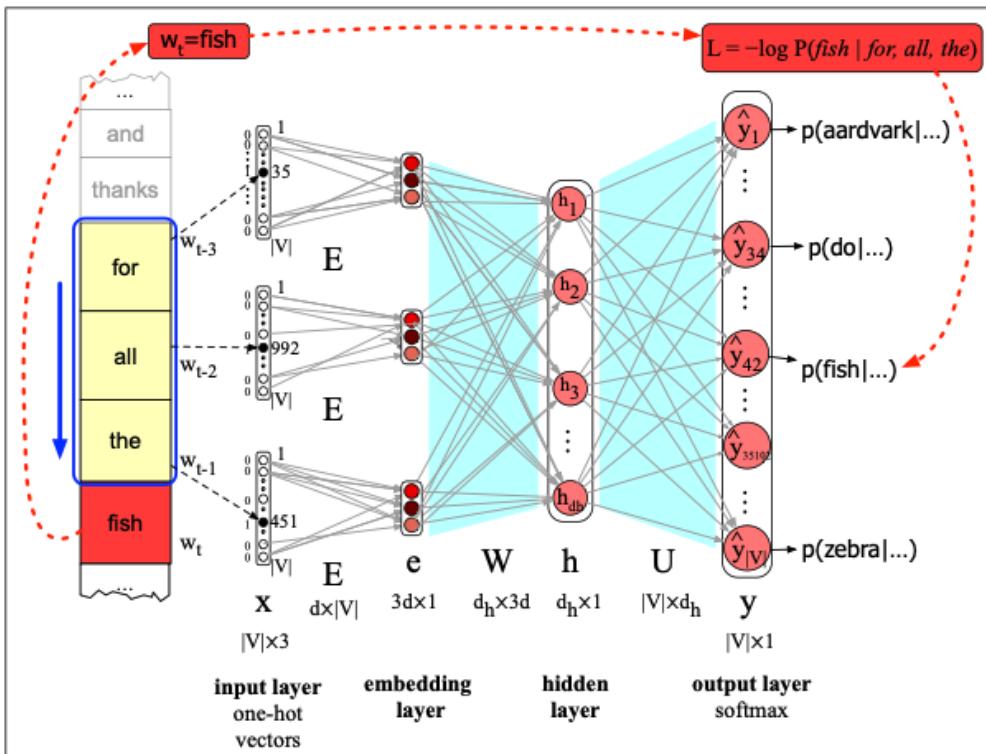


Figure 7.18 Learning all the way back to embeddings. Again, the embedding matrix \mathbf{E} is shared among the 3 context words.

Figura 2.16: Retomado de Jurafsky, et al (2024, Cap. 7, p.24) [JM23]

SENTIMENT ANALYSIS

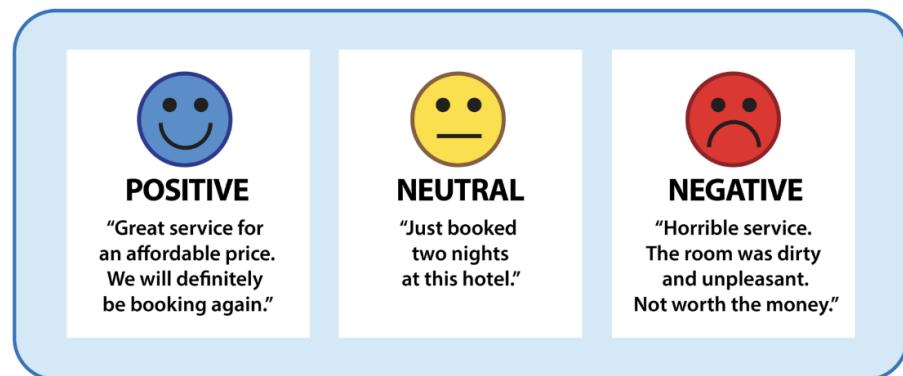


Figura 2.17: Sentiment

Nota: El NLP está separado del reconocimiento de voz, pero a menudo se usa junto con él, que busca analizar el lenguaje hablado en palabras, convirtiendo el sonido en texto y viceversa (ver, por ejmeplo: <https://medium.com/@kbdhunga/summarizing-youtube-videos-using-openai-whisper-gpt-3-5690c9a57>)

¿Por qué es importante el NLP?

Porque hoy día los agentes más sofisticados, como GPT-3 y GPT-4, se abrió para aplicaciones comerciales, pueden generar prosa sofisticada sobre una amplia variedad de temas, así como potentes chatbots capaces de mantener conversaciones coherentes.

Google utiliza el NLP para mejorar los resultados de sus motores de búsqueda y redes sociales como Facebook la utilizan para detectar y filtrar el discurso de odio.

El NLP es cada vez más sofisticado y será difícil dar seguimiento.

¿Para qué se utiliza el procesamiento del lenguaje natural (PLN)?

El análisis de sentimientos es el proceso de clasificar la intención emocional del texto.

La clasificación de toxicidad es una rama del análisis de sentimientos donde el objetivo no es sólo clasificar la intención hostil sino también clasificar

NAMED ENTITY RECOGNITION (NER) TAGGING

Andrew Yan-Tak Ng PERSON (Chinese NORP : 吳恩達; born 1976 DATE) is a British NORP -born American NORP computer scientist and technology entrepreneur focusing on machine learning and AI GPE . Ng was a co-founder and head of Google Brain ORG and was the former chief scientist at Baidu ORG , building the company's Artificial Intelligence Group ORG into a team of several thousand CARDINAL people.

Figura 2.18: Entity

categorías particulares como amenazas, insultos, obscenidades y odio hacia ciertas identidades.

La traducción automática automatiza la traducción entre diferentes idiomas.

El reconocimiento de entidades tiene como objetivo extraer entidades en un fragmento de texto en categorías predefinidas, como nombres personales, organizaciones, ubicaciones y cantidades.

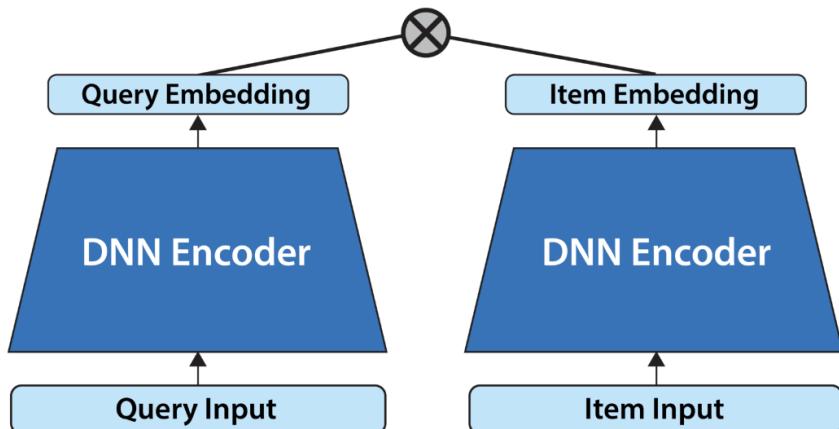
La detección de spam es un problema de clasificación binaria frecuente en PNL, donde el propósito es clasificar los correos electrónicos como spam o no. Los modelos de corrección de errores gramaticales codifican reglas gramaticales para corregir la gramática dentro del texto. El modelado de temas es una tarea de minería de textos no supervisada que toma un corpus de documentos y descubre temas abstractos dentro de ese corpus. La generación de texto, más formalmente conocida como generación de lenguaje natural, produce texto similar al texto escrito por humanos. La recuperación de información encuentra los documentos que son más relevantes para una consulta. Este es un problema al que se enfrenta todo sistema de búsqueda y recomendación.

2.4.2. Diseño de prompts

Usaremos el modelo chatGPT de OpenAI

Profundizaremos en más detalles sobre el formato y las entradas del chat

INFORMATION RETRIEVAL



A two-tower network creates a representation of an input query and a group of documents (or items) through two separate networks. Then it compares the representation of the query with that of the documents to find documents that are most relevant to the query.

Figura 2.19: Information

Ideas generales del Chat GPT 3.5 y GPT 4

En el desarrollo de grandes modelos de lenguaje (LLM), ha habido en términos generales dos tipos de LLM: LLM básicos (base LLM) y LLM ajustados por instrucción (instruction-tuned LLM).

1. Un base LLM ha sido entrenado para predecir la siguiente palabra basándose en datos de entrenamiento de texto, a menudo entrenado con una gran cantidad de datos de Internet y otras fuentes para determinar cuál es la siguiente palabra más probable a seguir. Por ejemplo, "había una vez un unicornio", ¿pueden completar esto?, ¿puede predecir la siguiente palabra?, digamos "que vivía en un bosque mágico con todos los amigos unicornios"
2. Un instruction-tuned LLM ha sido capacitado para seguir instrucciones. Por ejemplo, si le preguntará ¿cuál es la capital de Francia?, la respuesta es "la capital de Francia es París". Es decir, está afinado con entradas y salidas que son instrucciones

Entonces, cuando utilicemos un instruction-tuned LLM piensa en darle

Two Types of large language models (LLMs)

Base LLM

Predicts next word, based on text training data

Once upon a time, there was a unicorn that lived in a magical forest with all her unicorn friends

Instruction Tuned LLM

Tries to follow instructions

Fine-tune on instructions and good attempts at following those instructions.

RLHF: Reinforcement Learning with Human Feedback

Helpful, Honest, Harmless

What is the capital of France?
What is France's largest city?
What is France's population?
What is the currency of France?

What is the capital of France?
The capital of France is Paris.

Figura 2.20: LLM

instrucciones a otra persona, digamos a alguien que sea inteligente pero que no conozca los detalles de su tarea.

Entonces, cuando un LLM no funciona, a veces es porque las instrucciones no fueron lo suficientemente claras.

¿Cómo puede ayudarte el uso de Chat GPT en tu día a día?

Ir a GitHub

Bibliografía

- [Aba21] Alberto Abadie. «Using synthetic controls: Feasibility, data requirements, and methodological aspects». En: *Journal of Econometric Literature* (2021).
- [Ada20] Christopher P. Adams. *Learning Microeconomics with R*. Estados Unidos: CRC Press, 2020.
- [CT05] Colin Cameron y Pravin K. Trivedi. *Microeometrics: Methods and Applications*. Estados Unidos: Cambridge University Press, 2005.
- [Cha20] Mark Chang. *Artificial Intelligence for Drug Development, Precision Medicine, and Healthcare*. Estados Unidos: CRC Press Taylor & Francis Group, 2020.
- [Cun21] Scott Cunningham. *Causal inference: The mixtape*. Estados Unidos: Yale University Press, 2021.
- [Gre12] William Greene. *Econometric Analysis*. Estados Unidos: Prentice Hall, 2012.
- [HTF17] Trevor Hastie, Robert Tibshirani y Jerome Friedman. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*. LLC: Springer, 2017.
- [Jam+13] Gareth James et al. *An Introduction to Statistical Learning with Applications in R*. New York: Springer, 2013.
- [JM23] Dan Jurafsky y James H. Martin. *Speech and Language Processing*. New York: Stanford, 2023.
- [LM12] Richard J. Larsen y Morris L. Marx. *An Introduction to Mathematical Statistics and its Applications*. Estados Unidos: Prentice Hall, 2012.

- [Shm+19] Galit Shmueli et al. *Data Mining for Business Analytics: Concepts, Techniques and Applications in Python*. Estados Unidos: Wiley, 2019.
- [Woo10] Jeffrey M. Wooldridge. *Econometric Analysis of Cross Section and Panel Data*. Estados Unidos: The MIT Pres, 2010.