Las secciones son para no perderme durante la escritura

1. Marco teórico

1.1. Modelo relacional de base de datos

El *modelo relacional de base de datos*, consiste en cinco componentes:

1. Una colección de tipos escalares, pueden ser definidos por el sistema (INTEGER, CHAR, BOOLEAN, etc.) o por el usuario.
2. Un generador de tipos de relaciones y un intérprete para las relaciones mismas.
3. Estructuras para definir variables relacionales de los tipos generados.
4. Un operador para asignar valores de relación a dichas variables.
5. Una colección relacionalmente completa para obtener valores relacionales de otros valores relacionales mediante operadores relacionales genéricos.

Aunque tersa esta lista es útil para delimitar lo que el modelo relacional es y no es.

Se debe comenzar definiendo los *tipos*, ya que las relaciones se definen sobre ellos, son "en esencia un conjunto finito de valores nombrados—todos los valores posibles de alguna categoría específica(...)"Date 2012.

Los *atributos* son pares ordenados de combinaciones atributo-nombre/tipo-nombre y una *tupla* es un par ordenado de atributos. El modelo relacional también soporta varios tipos de *llaves*, que poseen las propiedades de unicidad, ninguna contiene dos tuplas distintas con el mismo valor e irreductibilidad, ningún subconjunto suyo es tiene unicidad.

	código	fecha	estado
Tuplas	MX01	29-07-99	1
	MX02	30-07-99	1
	MX03	31-07-99	0

Cuadro 1: Representación de una tabla de una base en datos, la fila superior muestra tres atributos distintos y *cada una* de las filas siguientes es una tupla.

La *restricción de integridad* (*constraint*) es una expresión booleana que debe evaluarse como verdadera. Los *constraints de tipo* definen los valores que constituyen

un tipo dado y los *constraints de base de datos* limitan los valores que pueden aparecer en cierta base de datos. Las bases de datos suelen tener múltiples constraints específicos, expresados en términos de sus relaciones, sin embargo, el modelo relacional incluye dos constraints genéricos, que aplican a cada base de datos:

- Regla de integridad de identidad: Las *llaves primarias* (PK) deben ser no nulas.
- Regla de integridad de referencia: Las *llaves foráneas* (FK) deben tener relación (si B referencia a A, A debe existir).

Las operaciones del modelo relacional están cimientadas en el *álgebra relacional*.

Utilizando operaciones primitivas del álgebra se producen nuevas relaciones que pueden manipularse también por medio de operaciones del álgebra mismo. Una secuencia de operaciones de álgebra relacional forma una *expresión de álgebra relacional* cuyo resultado es una relación que representa el resultado de una consulta (o solicitud de consulta) de base de datos.Elmasri y Navathe 2011 Estas operaciones se pueden clasificar en dos grupos, *operaciones de conjuntos* de la teoría de conjuntos matemáticos: *UNIÓN*, *INTERSECCIÓN*, *DIFERENCIA* y *PRODUCTO CARTESIANO* (*PRODUCTO CRUZADO*). El otro grupo consiste en operaciones específicas para bases de datos relacionales: *JUNTAR*, *SELECCIONAR* y *PROYECTAR*. Estas dos últimas, por operar con una sola relación, son también conocidas como *operaciones unarias*.

SELECCIONAR elige un subconjunto de tuplas de una relación que satisfacen la condición de selección

$$\sigma_{<\text{condición de selección}>}(\text{R}) \tag{1}$$

donde σ denota el operador SELECCIONAR, y la condición de selección es una expresión booleana especificada en los atributos de la relación R.

PROYECTAR produce una nueva relación de atributos y tuplas duplicadas

$$\pi_{<\text{lista de atributos}>}(\text{R}) \tag{2}$$

donde π es el denota el operador PROYECTAR y la lista de atributos es la sublista de atributos deseados de la relación R.

Las operaciones con dos relaciones reciben el nombre de *operaciones binarias*. Si las relaciones $R(A_1, A_2, \ldots, A_n)$ y $S(B_1, B_2, \ldots, B_n)$ son *compatibles con la unión* (tienen el mismo grado n y $dom(A_i) = dom(B_i)$ para $1 \leq i \leq n$) podemos usarlas para definir las siguientes operaciones binarias:

- UNIÓN: El resultado de esta operación se denota $R \cup S$, es una relación que incluye todas las tuplas que están en R, S o ambos, se eliminan duplicados.

- INTERSECCIÓN: El resultado de esta operación se denota $R \cap S$, es una relación que incluye todas las tuplas que están en ambos R y S.
- DIFERENCIA: El resultado de esta operación se denota $R - S$, es una relación que incluye todas las tuplas que están en ambos R pero no en S.

PRODUCTO CARTESIANO denotado $R \times S$ es la operación binaria que no requiere compatibilidad con la unión y produce un nuevo elemento al combinar cada miembro (tupla) de cada relación con cada otro miembro de la otra relación. El resultado de $R(A_1, A_2, \ldots, A_n)$ y $S(B_1, B_2, \ldots, B_m)$ es una relación Q con atributos $Q(A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m)$ (en ese orden) de grado $n + m$. El resultado Q tiene una tupla por cada combinación de tuplas de R y S.

JUNTAR $\bowtie$, es el operador utilizado para combinar tuplas relacionadas de dos relaciones en una sola tupla. Pertinente para procesar relaciones entre relaciones. Si tenemos dos relaciones $R(A_1, A_2, \ldots, A_n)$ y $S(B_1, B_2, \ldots, B_m)$ podemos escribir la operación JUNTAR como

$$R \bowtie_{<\text{condiciones de unión}>} S \tag{3}$$

el resultado es la relación Q con atributos $Q(A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m)$ (en ese orden). El resultado Q tiene una tupla por cada combinación de tuplas de R y S que satisfacen las condiciones de unión.

1.2. Mate

Una *variable aleatoria* (v.a.) es una función real $X : \Omega \mapsto \mathbb{R}$ tal que el conjunto $\{\omega \in \Omega : X(\omega) \in I\}$ es un evento de Ω para cada $I \subset \mathbb{R}$, en un espacio Ω hipotético. Se le considera *variable aleatoria discreta* (v.a.d.) cuando su rango de valores R_x es finito o contablemente infinito, mientras que una *variable aleatoria continua* (v.a.c.) puede tomar cualquier valor real en un intervalo.

La forma más natural de expresar la distribución de v.a.d.s es la *función de probabilidad*Blitzstein y Hwang 2019. Una v.a.d. X con $R_x = \{x_1, x_2 x_3, \ldots, x_n, \ldots\}$ tiene una función de distribución

$$\begin{gathered} f(x) = 0 \text{ para cada } x \notin R_x; \\ f(x) = P(X = x) \text{ para } x \in R_x \end{gathered} \tag{4}$$

para una v.a.c. X será una función no negativa real $f : \mathbb{R} \mapsto [0, \infty)$, es decir

$$P(X \in A) = \int_A f(x)dx \tag{5}$$

El *valor esperado* de una v.a.d. X con una función de probabilidad (4) es definida como

$$\mu = E(X) = \sum_{x \in R_x}^{\infty} x f(x), \tag{6}$$

siempre y cuando la serie converja absolutamente y es también llamado *media* de X, utilizada, similar a la media aritmética en estadísticas, para obtener el valor promedio entre observaciones.

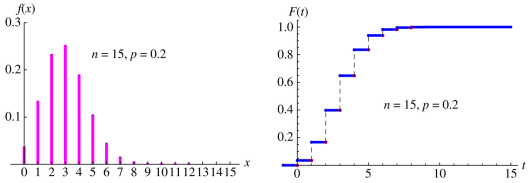
Para una v.a.c. X se define como

$$\mu = E(X) = \int_{-\infty}^{\infty} x f(x) dx \tag{7}$$

Para conocer la variabilidad de la distribución de cualquier v.a se utiliza la *varianza*, para X se define

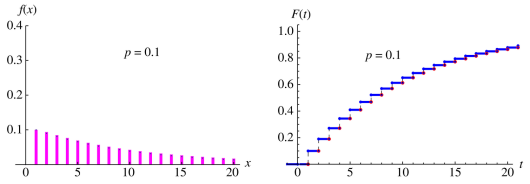
$$\sigma^2 = Var(X) = [(X - \mu)^2] \tag{8}$$

Figura 1: Existe diversidad de distribuciones para modelar v.a.s, a continuación se muestran las mas importantes de acuerdo a Balakrisnan, Koutras y PolitisBalakrishnan, Koutras y Politis 2020, siendo aptas para v.a.d.s hasta la distribución Normal, mientras que las v.a.c.s pueden ser modeladas a partir de la distribución Uniforme.



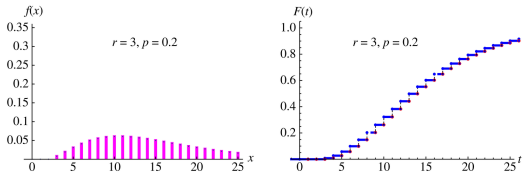
Binominal $b(n, p)$. Número de éxitos en n ensayos de Bernoulli independientes con la misma probabilidad de éxito p.

$$\begin{gathered} f(x = k) = \binom{n}{x} p^x p^{n-x}, \\ x = 0, 1, \ldots, n; \\ E(X) = np,\ Var(X) = npq \end{gathered} \tag{9}$$



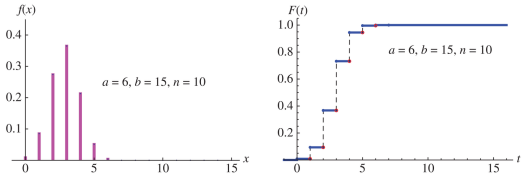
Geométrica $G(p)$. Número n de ensayos de Bernoulli independientes con la misma probabilidad de éxito p, hasta obtener el primer éxito.

$$\begin{gathered} f(x) = q^{x-1}, \\ x = 1, 2, \ldots, n; \\ E(X) = \frac{1}{p},\ Var(X) = \frac{q}{p^2} \end{gathered} \tag{10}$$



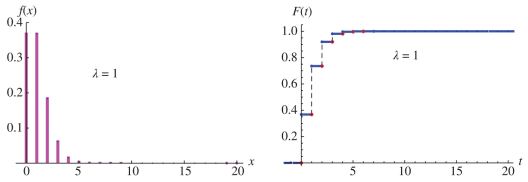
Negativa binominal $Nb(rp)$. Número n de ensayos de Bernoulli independientes con la misma probabilidad de éxito p, hasta obtener resultado número r.

$$\begin{gathered} f(x) = \binom{x-1}{r-1} p^r q^{x-r}, \\ r = r, r+1, r+2, \ldots, n; \\ E(X) = \frac{r}{p},\ Var(X) = \frac{rq}{p^2} \end{gathered} \tag{11}$$



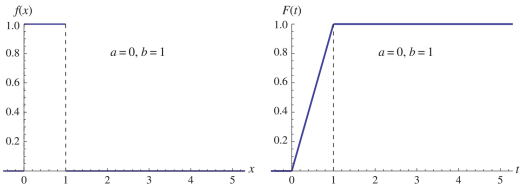
Hipergeométrica $h(n; a, b)$. Muestra aleatoria tamaño n de elementos tipo a no reemplazada de un universo con elementos tipo a y b.

$$\begin{gathered} f(x) = P(X = x) = \frac{\binom{a}{x}\binom{b}{n-x}}{\binom{a+b}{n}}; \\ E(X) = n \cdot \frac{a}{a+b}, \\ Var(X) = n \cdot \frac{a}{a+b} \cdot \frac{b}{a+b} \cdot \left(1 - \frac{n-1}{a+b-1}\right) \end{gathered} \tag{12}$$



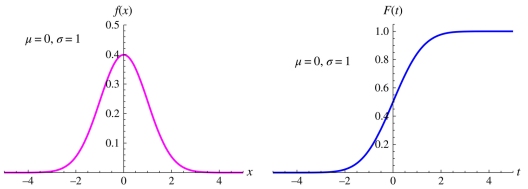
Poisson $\mathcal{P}(\lambda)$. Puede utilizarse con algún parámetro λ cuando la probabilidad de éxito tiende a cero ($p \to 0$) de forma que la media $E(X) = np$ converge en algún $\lambda > 0$.

$$\begin{gathered} f(x) = e^{-\lambda} \frac{\lambda^x}{x!}, \\ x = 0, 1, 2, \ldots; \\ E(X) = \lambda, Var(X) = \lambda \end{gathered} \tag{13}$$



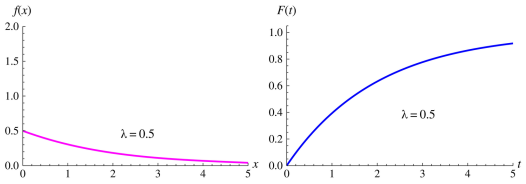
Uniforme $U[a, b]$. Útil para modelar situaciones en que todos los intervalos tengan la misma amplitud y probabilidad.

$$\begin{gathered} f(x) \begin{cases} \frac{1}{b-a}, & a \le x \le b, \\ \text{de otro modo}, & 0; \end{cases} \\ F(t) \begin{cases} 0, & t < 0, \\ \frac{t-a}{b-a}, & a \le t \le b, \\ 1, & t > b; \end{cases} \\ E(X) = \frac{a+b}{2}, \quad Var(X) = \frac{(b-a)^2}{12} \end{gathered} \tag{14}$$



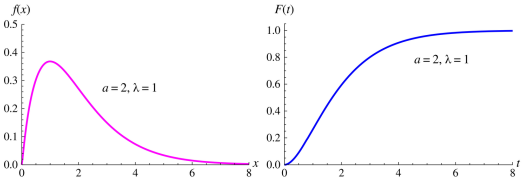
Normal $N(\mu, \sigma^2)$. La suma y el promedio de un gran número de observaciones para una variable X puede ser aproximada por la distribución normal, independientemente de su distribución original.

$$\begin{gathered} f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}, \\ -\infty < z < \infty; \\ E(X) = \mu,\ Var(X) = \sigma^2 \end{gathered} \tag{15}$$



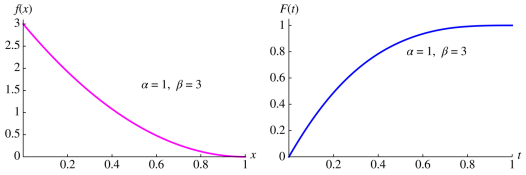
Exponencial $\epsilon(\lambda)$. Es considerada la análoga continua de la distribución geométrica y puede ser utilizada para modelar parte de la vida de un sujeto X.

$$\begin{gathered} f(x) = \begin{cases} \lambda^{-\lambda x},\ x \le 0, \\ 0,\ x < 0; \end{cases} \\ F(t) = \begin{cases} 0,\ t < 0, \\ 1 - e^{-\lambda t},\ x \ge 0; \end{cases} \\ E(X) = \frac{1}{\lambda},\ Var(X) = \frac{1}{\lambda^2} \end{gathered} \tag{16}$$



Gama. Generalización de la distribución *Erlang* cuando el parámetro n no es necesariamente un entero.

$$\begin{gathered} f(x) = \begin{cases} \frac{\lambda^a}{\Gamma(a)} x^{a-1} e^{-\lambda x},\ x \ge 0, \\ 0,\ x < 0, \end{cases} \\ \text{donde } \Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt \\ \text{es la función gama.} \\ E(X) = \frac{a}{\lambda},\ Var(X) = \frac{a}{\lambda^2} \end{gathered} \tag{17}$$



Beta. Ofrece modelos satisfactorios para v.a.c.s que toman valores entre dos puntos conocidos.

$$\begin{gathered} f(x) = \begin{cases} \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1},\ 0 < x < 1, \\ \text{de otro modo, } 0, \end{cases} \\ \text{donde } B(\alpha,\beta) = \\ \int_0^1 x^{\alpha-1}(1-x)^{\beta-1} dx = \\ \big(\Gamma(\alpha)\Gamma(\beta)\big) / \big(\Gamma(\alpha+\beta)\big) \\ \text{es la función beta.} \\ E(X) = \frac{a}{\lambda},\ Var(X) = \frac{a}{\lambda^2} \end{gathered} \tag{18}$$

1.3. AI, ML, NLP

Inteligencia artificial (IA) es *el esfuerzo por automatizar tareas intelectuales normalmente realizadas por humanos*(Chollet, 2018), de este campo general se desprenden el *aprendizaje maquinal* (AM) y *aprendizaje profundo* (AP).

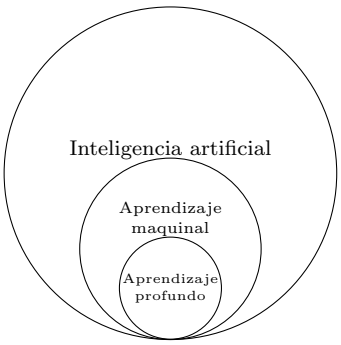


Figura 6: AP⊂AM⊂IA

Tom Michell (Mitchell, 1997) define aprendizaje maquinal como *un programa de computadora aprende de experiencia E con respecto a una tarea T y una medición de rendimiento P, si su rendimiento en T, medido por P, mejora con E.*

A diferencia del paradigma clásico de programación donde los humanos introducen datos y órdenes para procesarlos, un sistema de aprendizaje maquinal no se programada explícitamente, se introducen muchos ejemplos relevantes a una tarea (datos y respuestas esperadas) con los que es entrenado y si encuentra una estructura estadística en ellos, genera una regla para automatizar la tarea.

El *procesamiento de lenguaje natural* (PLN), es el conjunto de métodos para hacer accesible el lenguaje humano a las computadoras(Eisenstein, 2019). Toma conocimientos de muchas tradiciones intelectuales, como lingüística y teoría formal del lenguaje, autómatas y otras áreas computación, inteligencia artificial, aprendizaje maquinal y profundo, estadística, teoría de la información, fonética y fonología, estas dos últimas áreas son de particular utilidad para procesamiento de voz. Existen dos posturas opuestas sobre lo que la tarea principal del PLN debe ser:

- Entrenar sistemas de principio a fin para que transmuten texto sin procesar en cualquier estructura deseada.
- Transformar texto en una pila de estructuras lingüísticas de uso general que en teoría deben poder soportar cualquier aplicación.

En la actualidad no hay consenso y ambos tipos de sistemas se consideran viables, este proyecto se basará en el segundo paradigma.

Por lo general el PLN divide sus funciones en módulos para facilitar la reutilización de algoritmos genéricos en diversas tareas y modelos, dos de los módulos básicos son *búsqueda* y *aprendizaje* con los que se puede resolver muchos problemas que tienen la forma matemática

$$\hat{y} = argmax\Psi(x, y; 0), \quad y \in Y(x) \tag{19}$$

donde,

- x es la entrada, un elemento de un conjunto X.
- y es el resultado, un elemento de un conjunto Y.
- Ψ es una función de puntuación (también conocida como *modelo*), que va desde el conjunto $X \times Y$ hasta los números reales.
- $\emptyset$ es el vector de parámetros para Ψ.
- $\hat{y}$ es el resultado previsto, que es elegido para maximizar la función de puntuación.

El módulo de búsqueda se encarga de computar el $argmax$ de la función Ψ, es decir, encuentra el resultado $\hat{y}$ con la mejor puntuación con respecto a la entrada x. El módulo de aprendizaje encuentra los parámetros θ por medio del procesamiento de grandes conjuntos de datos de ejemplos etiquetados $\{(x^i, y^i)\}_{i=1}^{N}$.

Un método lineal de clasificación de texto común es la *bolsa de palabras*, comienza por asignar etiquetas $y \in Y$ donde Y son todas las posibles etiquetas. Se utilizan vectores columna y la fórmula 19 y puede modelarse con diversas distribuciones (figura 1).

Para muchas tareas, las características léxicas (palabras) pierden sentido en aislamiento, por lo que históricamente el PLN se ha enfocado en la clasificación lineal, recientemente algunas tareas pueden resolverse con clasificadores no lineales, es decir, por medio de redes neuronales (aprendizaje profundo).

El aprendizaje profundo sigue el paradigma del aprendizaje maquinal, sus entradas son datos y ejemplos de resultados esperados y se mide si el algoritmo está haciendo un buen trabajo mientras *aprende*. A diferencia del aprendizaje maquinal, se hace énfasis en aprender en capas sucesivas de representaciones cada vez más sucesivas. "Profundidad" entonces refiere a cuántas capas contribuyen a un modelo.

2. Objetivos

Comenzar desde cero cada proyecto es ineficiente. (esto es temporal)

2.1. General

Diseñar y desarrollar una
plataforma/framework de ML para NLP
reutilizable y/o de uso general/ ¿para casos de uso similares?

2.2. Particulares

Revisé algunas tesis para darme la idea, necesito saber un poco más del tema para desarrollar esta parte, creo que sería algo así:

1. Investigación
2. Análisis info
3. Diseño/desarrollo
4. Comprobación

##

Al integrar
/tecnologías/técnicas/modelos/librerías/frameworks/
de RDB, ML, NLP, ¿transfer learning?, ¿deep learning?,...
se de puede /diseñar/desarrollar/, ¿e implementar? una
/plataforma/framework/
/genérica/normalizada/universal/reutilizable/estandarizada/compatible
/con/en/para/ /casos de uso similares/diversos casos de uso/
(/reduciendo tiempo/ahorrando recursos/.) estas oración se sale del scope

LO NUEVO Haciendo uso de las ciencias de la computación, las herramientas matemáticas de estadística y probabilidad, y métodos de aprendizaje autónomo se busca obtener información cuantitativa de textos provenientes de redes sociales, cadenas noticiosas y audio de programas de capacitación, para inferir posiciones, tendencias, comportamientos o razones de grupos sociales, considerando un ciclo de clasificación, estimación, detección y comprobación.

Particulares Desarrollar un modelo de base de datos que permita la captura de categorías para un determinado problema, los elementos de identificación de cada categoría, el origen de la información y su correlación. Construir una estructura de datos que capte la estimación o valores esperados para el procesamiento de textos Elaborar un sistema de objetos para el soporte de los elementos de aprendizaje autónomo Generar los elementos de captura de textos para su almacenamiento y procesamiento Elaborar un modelo estadístico que permita comprobar las estimaciones a partir de los datos y en consecuencia realizar un ajuste en los parámetros usados para el aprendizaje autónomo Producir los reportes con un análisis estadístico que faciliten la interpretación de resultados y den pauta para la obtención del conocimiento de interés.

3. Metodologías

4. Referencias

Balakrishnan, N., Markos V. Koutras y Konstantinos G. Politis (2020). *Introduction to probability: models and applications [Introducción a la probabilidad: modelos y aplicaciones]*. Hoboken: John Wiley & Sons, Inc. ISBN: 9781118123348.

Blitzstein, Joseph K. y Jessica Hwang (2019). *Introduction to Probability [Introduccién a la probabilidad]*. 2.ª ed. Texts in Statistical Science [Textos en ciencia estadistica]. Florida: CRC Press. ISBN: 9781138369917.

Chollet, François (2018). *Machine Learning With Python [Machine Learning]*. New York: Manning Publications Co. ISBN: 9781617294433.

Date, C. J. (2012). *SQL and Relational Theory: How to Write Accurate SQL Code [SQL y teoria relacional: Cómo escribir código SQL correcto]*. 2.ª ed. California: 0'Reilly Media. ISBN: 9781449316402.

Eisenstein, Jacob (2019). *Introduction to natural language processing [Introducción al procesamiento de lenguaje natural]*. Adaptive computation and machine learning [Computación adaptativa y aprendizaje maquinal]. Cambridge: MIT Press. ISBN: 9780262042840.

Elmasri, Ramez y Shamkant B. Navathe (2011). *Fundamentals of database systems [Fundamentos de sistems de bases de datos]*. 6.ª ed. Boston: Addison-Wesley. ISBN: 9780136086208.

Mitchell, Tom (1997). *Machine learning [Aprendizaje maquinal]*. New York: MacGraw - Hill. ISBN: 0070428077.