UNIVERSIDAD
COMPLUTENSE
MADRID

Minería de datos

Sistemas de Gestión de Datos y de la Información
Enrique Martín - emartinm@ucm.es
Máster en Ingeniería Informática
Fac. Informática

Contenidos

1) Introducción
2) Terminología
3) Técnicas de aprendizaje automático
4) Extracción y transformación de atributos
5) Métricas de calidad de los modelos generados
6) Bibliografía

Introducción

Introducción

- Como ya conocemos, en el mundo actual se genera una gran cantidad de datos:
  - Redes sociales
  - Satélites de agencias espaciales
  - Páginas web en Internet
  - Grandes proyectos científicos: LHC, genoma, etc.
  - Sensores

Introducción

- Es importante **obtener conocimiento** de estos datos en bruto.
- La información que querríamos extraer son **patrones** que nos sirvan para entender mejor una determinada realidad. Ej:
  - **Si** un cliente compra leche y fruta, **entonces** compra cereales con probabilidad 0,6.

Introducción

- Estos patrones, obtenidos a partir de múltiples instancias, sirven para **comprender mejor la realidad**, lo que lleva a poder desarrollar teorías que expliquen y predigan mejor los fenómenos observados. Ejemplos:
  - Modelos climatológicos
  - Modelos económicos
  - Modelos de comportamiento de usuarios

Introducción

- La **minería de datos** *(data mining)* es el proceso automático (o semiautomático) que intenta descubrir patrones en grandes volúmenes de datos.
- Por otro lado, el **aprendizaje automático** *(machine learning)* es la rama de la informática que estudia los sistemas que pueden **aprender** a partir de datos.

Introducción

- En principio son disciplinas diferentes, pero están estrechamente relacionadas:

La minería de datos utiliza técnicas provenientes del aprendizaje automático (entre otras) para encontrar patrones en grandes volúmenes de datos.

Ejemplo de minería de datos

Sexo	Altura	Peso	Exp.	Act.
H	1,55	45	A+	Fútbol
H	1,67	58	C	Fútbol
M	1,45	45	B	Fútbol
M	1,58	50	A+	Pádel
H	1,20	40	B	Fútbol
M	1,80	60	A	Pádel
M	1,35	39	B	Fútbol
...	...	...	...	...

- Imaginemos que tenemos datos sobre los alumnos de un colegio (sexo, altura, peso y nota media del expediente) y qué actividad extraescolar han elegido.

Ejemplo de minería de datos

- Un ejemplo de minería de datos sería obtener reglas que nos permitan representar el contenido de la tabla:
  - **Si** sexo=H **entonces** act=fútbol
  - **Si** sexo=M y exp<A **entonces** act=fútbol
  - **Si** sexo=M y exp>=A **entonces** act=pádel
- Hemos encontrado patrones que se cumplen en nuestro conjunto de datos

Terminología

Instancias

Sexo	Altura	Peso	Exp.	Act.
H	1,55	45	A+	Fútbol
H	1,67	58	C	Fútbol
M	1,45	45	B	Fútbol
M	1,58	50	A+	Pádel
H	1,20	40	B	Fútbol
M	1,80	60	A	Pádel
M	1,35	39	B	Fútbol
...	...	...	...	...

- Un conjunto de datos está compuesto de ejemplos, llamados **instancias**.
- Cada fila de la tabla será una instancia.

Atributos

Sexo	Altura	Peso	Exp.	Act.
H	1,55	45	A+	Fútbol
H	1,67	58	C	Fútbol
M	1,45	45	B	Fútbol
M	1,58	50	A+	Pádel
H	1,20	40	B	Fútbol
M	1,80	60	A	Pádel
M	1,35	39	B	Fútbol
...	...	...	...	...

- Cada instancia consta de una serie de valores para unos **atributos**.
- Cada columna de la tabla corresponde con un atributo.

Clase

Sexo	Altura	Peso	Exp.	Act.
H	1,55	45	A+	Fútbol
H	1,67	58	C	Fútbol
M	1,45	45	B	Fútbol
M	1,58	50	A+	Pádel
H	1,20	40	B	Fútbol
M	1,80	60	A	Pádel
M	1,35	39	B	Fútbol
...	...	...	...	...

- En algunos casos hay un atributo especial, llamado **clase**, y la tarea es predecir el valor de clase para instancias nuevas.
- En este ejemplo la clase es la actividad.

Tipos de atributos

- Los atributos pertenecen a dos tipos principales: **categóricos** y **continuos**.
- Los atributos categóricos toman valores de un conjunto finito de valores posibles. Ejemplos:
  - (**Nominal**) Tiempo: soleado, nublado, lluvioso
  - (**Ordinal**) Valoración: malo < regular < bueno
- Los atributos continuos toman valores enteros o reales. Ejemplos: número de hijos (0,1,2...), temperatura (0ºC, 35ºF)

Tipos de aprendizaje

- A los datos que poseen el atributo especial de clase se les llama **datos etiquetados**.
- La minería de datos sobre datos etiquetados se conoce como **aprendizaje supervisado**.
- Si la clase es de tipo categórico, la tarea se conoce como **clasificación**, si la clase es de tipo continuo, la tarea se conoce como **regresión**.
- El ejemplo anterior se corresponde con clasificación (la clase es categórica).

Tipos de aprendizaje

- A los datos que no poseen el atributo especial de clase se les llama **datos no etiquetados**.
- La minería de datos sobre datos no etiquetados se conoce como **aprendizaje no supervisado**.
- Dentro de este tipo de aprendizaje destaca:
  - La generación de reglas de asociación que vinculan los valores de unos atributos con otros.
  - La obtención de grupos de instancias comunes *(clustering)*

Conjuntos de datos

- En minería de datos, para un problema dado se suelen distinguir 2 conjuntos de datos (*holdout method*):
  - El **conjunto de entrenamiento** *(training set)*. Sirve para entrenar al algoritmo de aprendizaje automático.
  - El **conjunto de test** *(test set)*. Son instancias **diferentes al conjunto de entrenamiento.** Sirve únicamente para medir la calidad del modelo final.
- Cuando existen hiper-parámetros o queremos elegir entre varios algoritmos, se puede considerar un tercer **conjunto de validación** (*validation set*) para escoger la mejor elección.

Sesgo en los datos

- Al usar aprendizaje automático es importante no tener ninguna clase **sobrerrepresentada**
- Si el 98% de las instancias del conjunto de entrenamiento tienen clase **positivo**, el aprendizaje más usual será ***“todo es positivo”*** *(y cometerá muy pocos fallos)*
- Esto puede ser importante también en los demás atributos que no son la clase

Sesgo en los datos

- Existen dos técnicas principales para solucionar el problema del sesgo:
  - **Sobremuestreo (*oversampling*)** aleatorio: replicar instancias con clases en minoría
  - ***Inframuestreo (undersampling)*** aleatorio: eliminar instancias con clase en mayoría
- Ambas técnicas se pueden combinar: eliminar algunas instancias de la clase en mayoría y replicar algunas instancias de la clase en minoría

Preparación de los datos

- Una tarea importante que hay que realizar antes de la minería de datos es la **preparación de los datos**.
- Esta tarea trata de eliminar anomalías en las instancias del conjunto de datos:
  - El valor de un atributo categórico está mal escrito
  - Falta el valor de un atributo
  - Un atributo continuo toma únicamente 5 valores diferenciados → convertir en atributo categórico
  - Un atributo toma siempre el mismo valor
  - Hay valores extremos para un atributo

Preparación de los datos

- Aunque sea un asunto importante, en el resto del tema consideraremos que todos los datos han sido previamente **preparados**.

Técnicas de aprendizaje automático

Minería de datos: visión global

- Aprendizaje supervisado
  - Clasificación
    - Clasificación de una instancia nueva (k-NN, naive bayes)
    - Reglas de clasificación
    - Árboles de clasificación
  - Regresión
- Aprendizaje no supervisado
  - Reglas de asociación
  - *Clustering*

Clasificación de instancia nueva

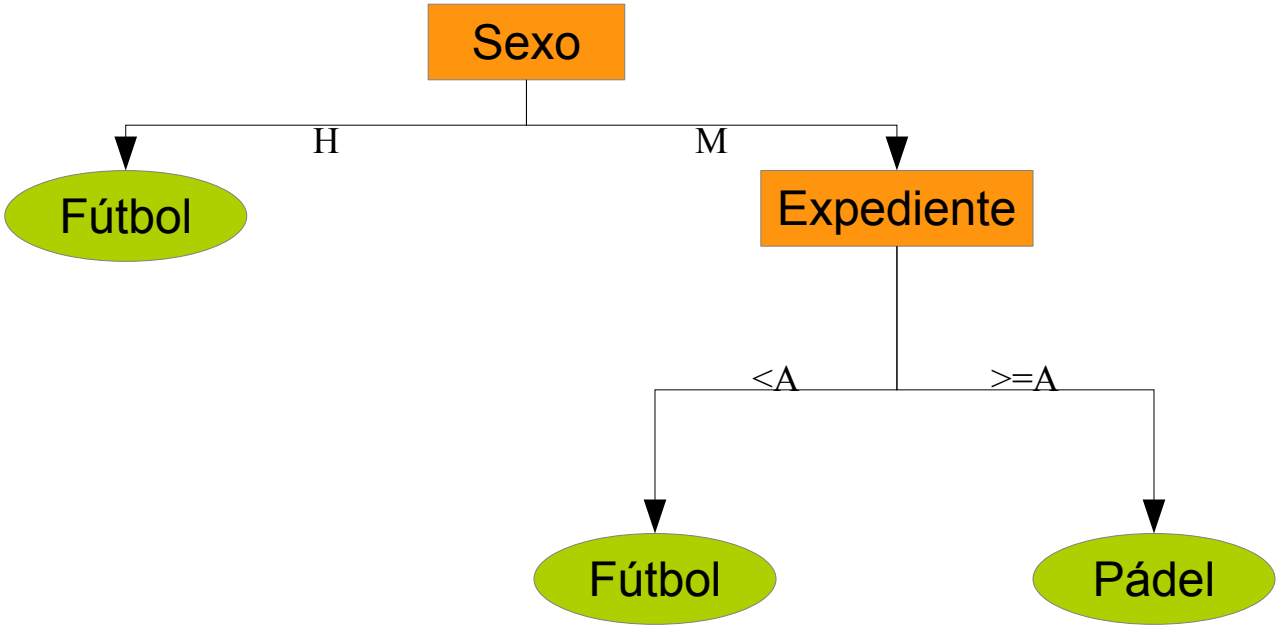
- Este método acepta un conjunto de datos y una instancia nueva.
- A la instancia nueva se le predice una clase, basándose en las instancias que aparecen en el conjunto de clases.
- Únicamente clasifica una instancia nueva, **no representa el conocimiento** de ninguna forma.

Reglas de clasificación

- Dado un conjunto de datos, genera una serie de reglas que sirven para predecir la clase de las nuevas instancias. Ejemplo:
  - **Si** sexo=H **entonces** act=fútbol
  - **Si** sexo=M y exp<A **entonces** act=fútbol
  - **Si** sexo=M y exp>=A **entonces** act=pádel
- Estas reglas sirven para clasificar fácilmente nuevas instancias, pero también representan el conocimiento extraído de forma sencilla.

Árbol de clasificación

- Dado un conjunto de datos, genera un árbol de clasificación para predecir la clase de las nuevas instancias. Ejemplo:

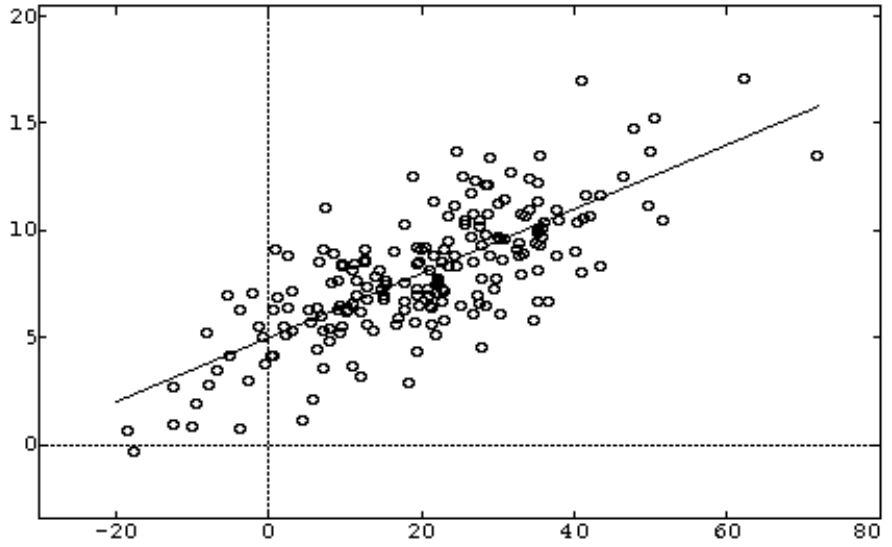


Árbol de clasificación

- Al igual que las reglas de clasificación, el árbol de clasificación sirve para clasificar fácilmente nuevas instancias, pero también **representan el conocimiento extraído** de forma sencilla (en este caso de manera gráfica)
- A partir de un árbol de clasificación se pueden crear las reglas de clasificación. Únicamente hay que recoger las condiciones por cada rama hasta llegar a una hoja.

Regresión

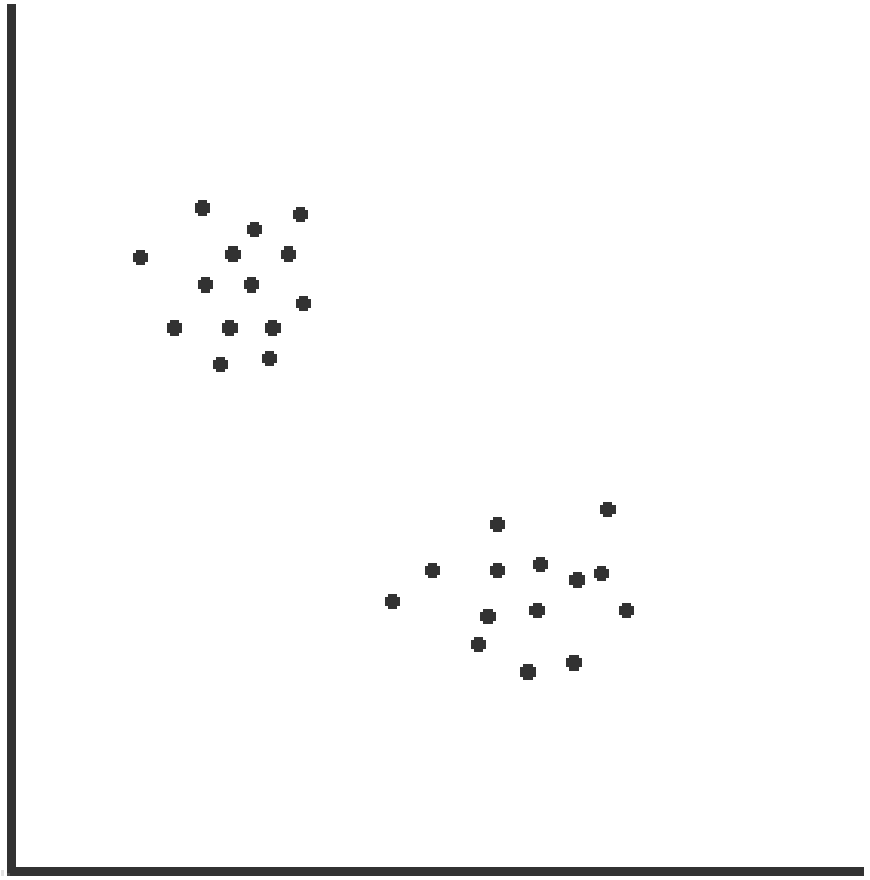
- En datos etiquetados con atributos a1,...,an **continuos**, querremos predecir el valor continuo de la clase para una instancia dada.



Reglas de asociación

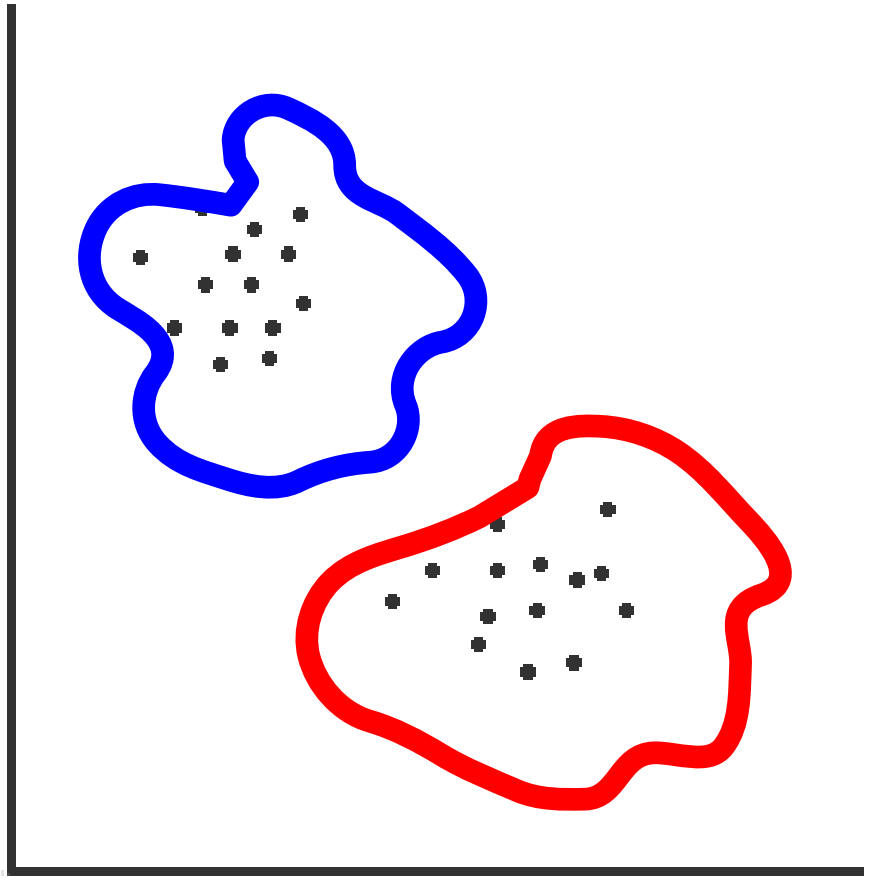
- En datos no etiquetados (no existe un atributo especial), querríamos encontrar relaciones entre distintos atributos.
- En principio nos interesa cualquier regla que muestre relación entre atributos.
- Para cada regla, tendremos valores que nos indicarán su calidad: **confianza**, **soporte** y **completitud**.

Clustering



- Considerando unos datos no etiquetados, la tarea de *clustering* (agrupado) trata de encontrar conjuntos de instancias similares.

Clustering



- En este caso tenemos 2 conjuntos claramente separados

Extracción y transformación de atributos

Atributos categóricos

- Muchas técnicas de aprendizaje automático únicamente manejan **atributos continuos** representados como números reales.
- Si nuestro conjunto de datos contiene atributos categóricos (normalmente cadenas de texto) debemos transformarlos primero a atributos continuos.
- Podemos destacar dos técnicas:
  - Ordinal encoder
  - One-hot encoder

Ordinal encoder

- Consiste en transformar cada valor categórico en un **valor real**
- De esta manera si un atributo categórico trivaluado **color** puede tomar valores {R,G,B} lo codificaríamos como:
  - R → 0
  - G → 1
  - B → 2

Ordinal encoder

- Sin embargo, esta codificación {R → 0, G → 1, B → 2} implica que **R está más alejado de B que de G**.
- Algunos algoritmos basados en distancia podrán aprender basándose en esta *noción de lejanía*, que es algo completamente **artificial** introducido por nuestra codificación.

Ordinal encoder

- Si el atributo categórico es ordinal y además las categorías están **uniformemente separadas**, *ordinal encoder* puede ser adecuado.
- Por ejemplo, {Muy mala → 0, Mala → 1, Regular → 2, Buena → 3, Muy buena: 4}
- Ahora la distancia entre “*muy mala*” y “*muy buena*” es máxima (4), y la distancia entre “*muy mala*” y “*regular*” (2) es la misma que entre “*regular*” y “*muy buena*” (2)
- La codificación expresaría las mismas ideas de distancia que tenemos en mente.

One-hot encoder

- Se crea un **nuevo atributo binario** por cada valor diferente del atributo categórico:

Color	→	ColorR	ColorG	ColorB
R	→	1	0	0
G	→	0	1	0
B	→	0	0	1

- La codificación *one-hot* no presenta problemas con las distancias, y suele ser la mejor opción

Manejo de atributos textuales

- Para poder utilizar atributos textuales en aprendizaje automático es necesario transformarlos a **vectores numéricos**
- El primer paso es separar el texto en **listas de palabras** (*tokens*), descubriendo a la vez el **vocabulario**
- Adicionalmente se puede:
  - eliminar ***stop words*** que no aportan significado
  - aplicar ***stemming*** o **lematización** para quedarse con la raíz o lema correspondiente (evitar formas flexionadas como plurales, tiempos verbales, etc.)

Manejo de atributos textuales

- Una vez tenemos una lista de palabras debemos transformarla en un **vector de números reales**
- **Podemos destacar dos técnicas:**
  - **Bag of words** (bolsa de palabras)
  - **Word2Vec**

Bag of words

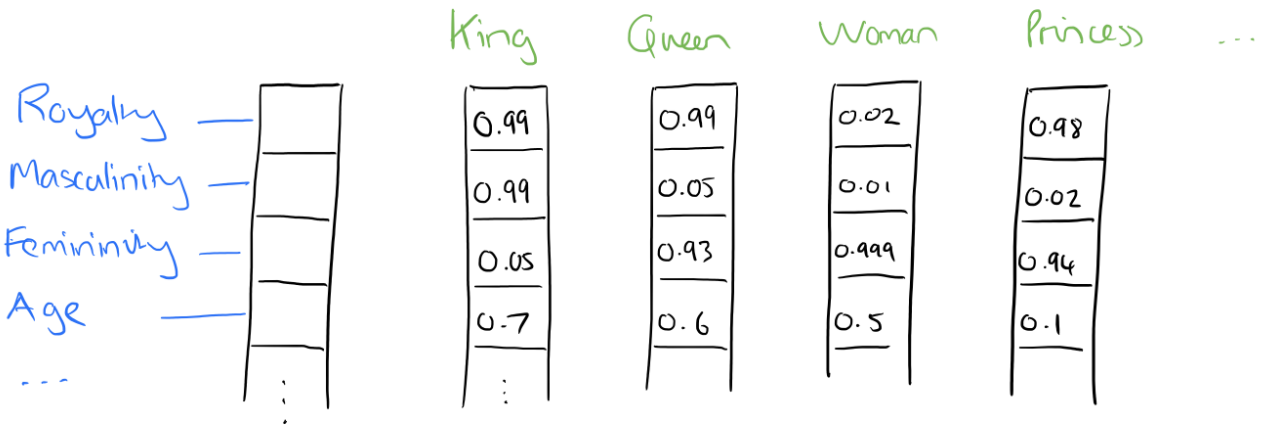
- Cada texto se representa como un **vector de apariciones de palabras**, de manera similar a lo que hemos visto en recuperación de información.
- Cada **palabra** se mapea a una **posición** del vector.
- La información asociada a cada palabra puede tener más o menos **detalle**:
  - si aparece o no: (1,1,0,0,1)
  - el número de veces que aparece: (5,1,0,0,3)
  - algo más elaborado como TF-IDF: (2.1, 1.1, 0, 0, 1.6)

Bag of words

- Un problema de la técnica *bag of words* es que **pierde** completamente la **posición** de las palabras y sus **relaciones**: únicamente expresa su aparición
- De esta manera textos como *“rey fuerte”* o *“reina enérgica”* se representarán de manera muy diferente aunque expresan ideas muy cercanas: “monarquía poderosa”
- Esta pérdida de información puede ser una **limitación** grave en algunos contextos

Word2vec

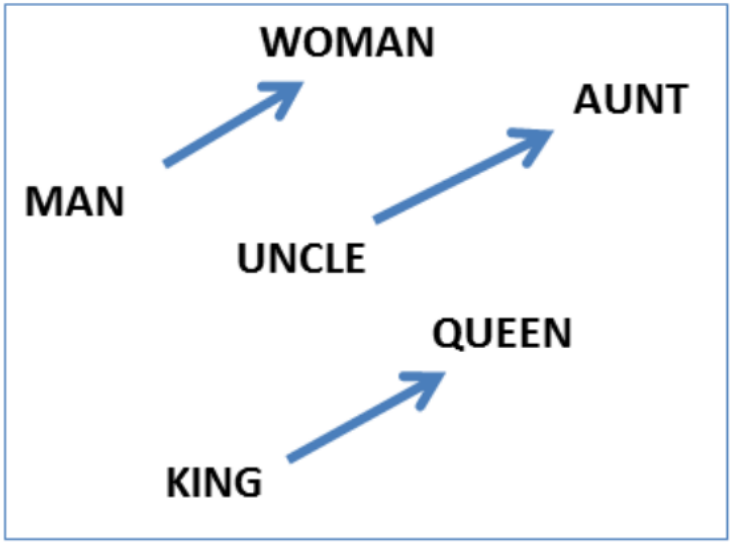
- **Intuición**: Word2vec es una técnica que representa la **connotación** de cada palabra en distintas **dimensiones**
- Cada palabra se representa como un vector en ese espacio de dimensiones:



Fuente: https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/

Word2vec

- Esta representación nos permite disponer de **relaciones vectoriales** entre los vectores de las **palabras relacionadas**.
- Por ejemplo, el femenino de una palabra será el mismo vector pero aumentando en la dimensión de *feminidad* y disminuyendo en la de *masculinidad*

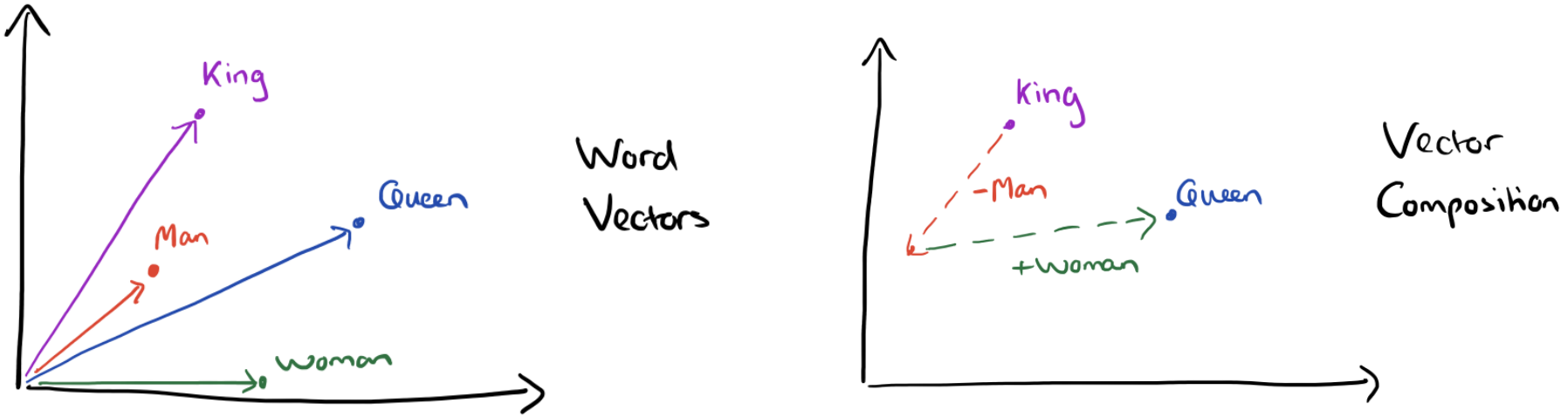


Fuente: https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/

Word2vec

- Este representación nos permite incluso un **razonamiento composicional** sobre las palabras:

Queen = (King – Man) + Woman



Fuente: https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/

Word2vec

- Existen distintas técnicas para aprender los vectores de cada palabra:
  - Continuous Bag-of-Words model (CBOW)
  - Continuous Skip-gram model
- Podéis encontrar más información en
  - *The amazing power of word vectors*, Adrian Coyler (2016) *https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/*
  - *word2vec Parameter Learning Explained*, Xin Rong (2016) https://arxiv.org/pdf/1411.2738v3.pdf
  - *Efficient Estimation of Word Representations in Vector Space*, Tomas Mikolov et at. (2013) https://arxiv.org/pdf/1301.3781.pdf

Word2vec en textos completos

- Podemos generar la **representación de un texto** completo como la **media de los vectores** de todas las palabras (este es el enfoque usado en **SparkML**)
- También existen extensiones de Word2vec para inferir directamente la representación de los documentos (**doc2vec**):
  - *Distributed Representations of Sentences and Documents*, Quoc Le & Tomas Mikolov (2014) https://arxiv.org/pdf/1405.4053.pdf

Escalado

- Otro aspecto a tener en cuenta es la **escala** de los atributos continuos, ya que también puede afectar a la **noción de *distancia***
- Ejemplo: A = (50 km, 2 puertas)
  B = (75 km, 5 puertas)
  C = (80 km, 2 puertas)
- ¿Qué instancia está más cerca de C: A o B?
  dist(A,C) = 30
  dist(B,C) = **5.8**
- La instancia más cercana a C es **B**.

Escalado

- Qué ocurre si mido el kilometraje en bloques que 1.000 kilómetros (megametros Mm)
  Ejemplo:
  A = (0,05 Mm, 2 puertas)
  B = (0,075 Mm, 5 puertas)
  C = (0,08 Mm, 2 puertas)
- ¿Qué instancia está más cerca de C: A o B?
  dist(A,C) = **0.03**
  dist(B,C) = 3
- Ahora la instancia más cercana a C es **A**.

Escalado

- Para evitar la influencia de las unidades de medida y no conceder más peso (inadvertidamente) a alguno de los atributos, se realiza un proceso de **escalado**
- Existen varios tipos, uno de los más simples es el **escalado MinMax**:

$$escalado_{MinMax}(a) = \frac{a - min}{max - min}$$

Escalado MinMax

- A = (50 km, 2 puertas)
  B = (75 km, 5 puertas)
  C = (80 km, 2 puertas)
- Kilometraje: min=50, max=80
- Número de puertas: min=2, max=5
- Resultado del escalado:
  - A = (0, 0)
  - B = (0.83, 1)
  - C = (1, 0)

Métricas de calidad de los modelos generados

Métricas de calidad

- Una vez tenemos un **modelo**, o si estamos eligiendo entre diferentes modelos (usando el *validation set*) debemos medir la **calidad de sus resultados**
- Métricas de calidad
  - **Clasificación**: matriz de confusión, exactitud, sensibilidad, especificidad, F_1...
  - **Regresión**: MSE, RMSE, MAE...
  - **Reglas de asociación:** confianza, soporte y completitud
  - ***Clustering****: Silhouette coefficent*, índice *Calinski-Harabasz*

Matriz de confusión

- Inicialmente para **clasificación binaria**
- Basada en 4 medidas:
  - **TP** (*true positive*): número de instancias clasificadas como positivas (P) que realmente lo son (T)
  - **FP** (*false positive*): número de instancias clasificadas como positivas (P) que realmente no lo son (F)
  - **TN** (*true negative*): número de instancias clasificadas como negativas (N) que realmente lo son (T)
  - **FN** (*false negative*): número de instancias clasificadas como negativas (N) que realmente no lo son (F)

Matriz de confusión

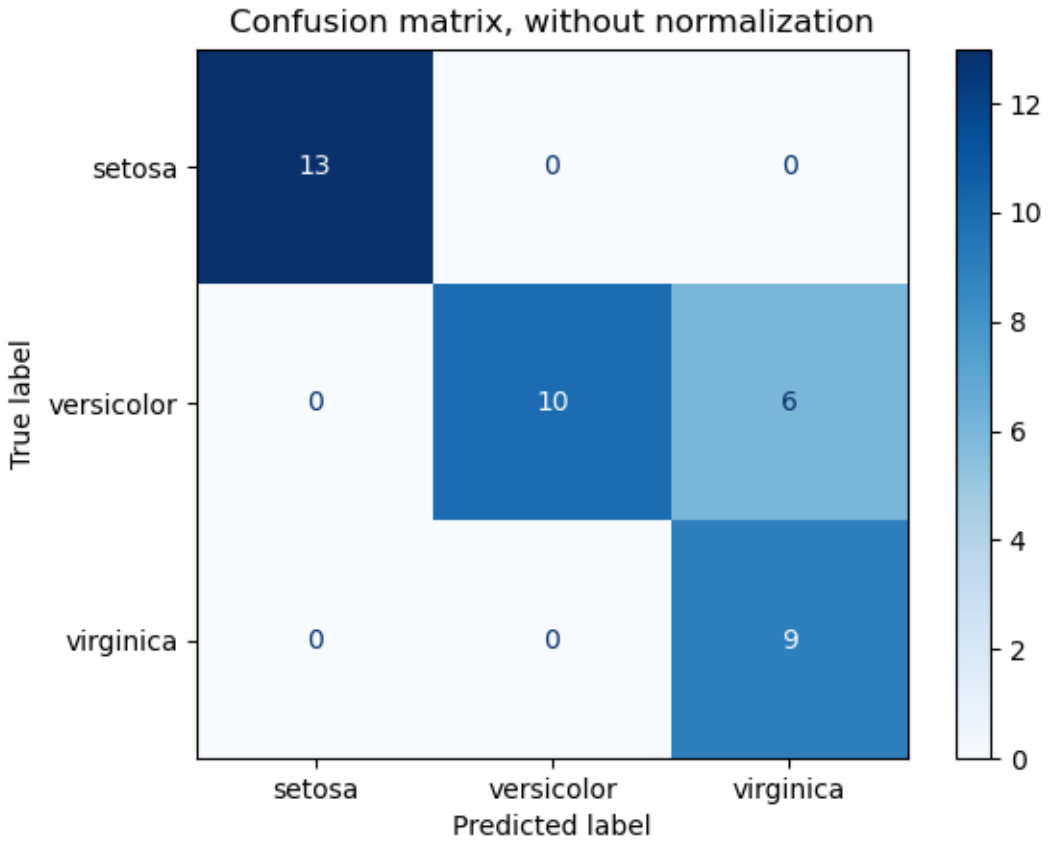
- Estas 4 medidas se representan en una matriz 2x2

		Predicción	
		Positivo	Negativo
Valor real	Positivo	TP	FN
	Negativo	FP	TN

- Idealmente queremos que **FN** y **FP** sean lo más cercano a 0

Matriz de confusión

- La matriz de confusión se puede generalizar a clasificación múltiple:



Fuente: https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

Exactitud

- La exactitud (*accuracy*) es la proporción **de instancias correctamente clasificadas** entre **todas las instancias probadas**

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

Sensibilidad

- La sensibilidad es la **proporción** de **instancias positivas correctamente clasificadas** entre el **total de instancias positivas**
- También llamado *true positive rate* (TPR) o *recall*

$$TPR = \frac{TP}{TP + FN}$$

Especificidad

- La especificidad es la **proporción** de **instancias negativas correctamente clasificadas** entre el **total de instancias negativas**
- También llamado *true negative rate* (TNR)

$$TNR = \frac{TN}{TN + FP}$$

Precisión

- La precisión (*precision*) es la **proporción** de **instancias positivas correctamente clasificadas** entre el **total de instancias clasificadas como positivas**
- También llamado *positive predictive value* (PPV)

$$PPV = \frac{TP}{TP + FP}$$

Valor-F (F-score)

- El valor-F (F_1) es una medida numérica que combina la **precisión y la sensibilidad** de un clasificador usando la media armónica

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

- **F_1** asigna el mismo peso a la precisión y a la sensibilidad. Si queremos dar más peso a la precisión o la sensibilidad usaremos **F_β**

Regresión

- Para medir la calidad de un modelo de regresión, acumulamos los errores producidos por cada instancia
  - Error cuadrático medio (MSE)

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

  - Error absoluto medio (MAE)

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

  - Raíz del error cuadrático medio (RMSE)

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

Bibliografía

Bibliografía

- Data Mining: Concepts and Techniques. *Jiawei Hai, Micheline Kamber*. Morgan Kauffmann (2001).
- Principles of Data Mining, second edition. *Max Bramer*. Springer (2013).
- Scikit-learn User Guide https://scikit-learn.org/stable/user_guide.html