TDVI: Inteligencia Artificial

Ingeniería de Atributos

UTDT - LTD

Estructura de la clase

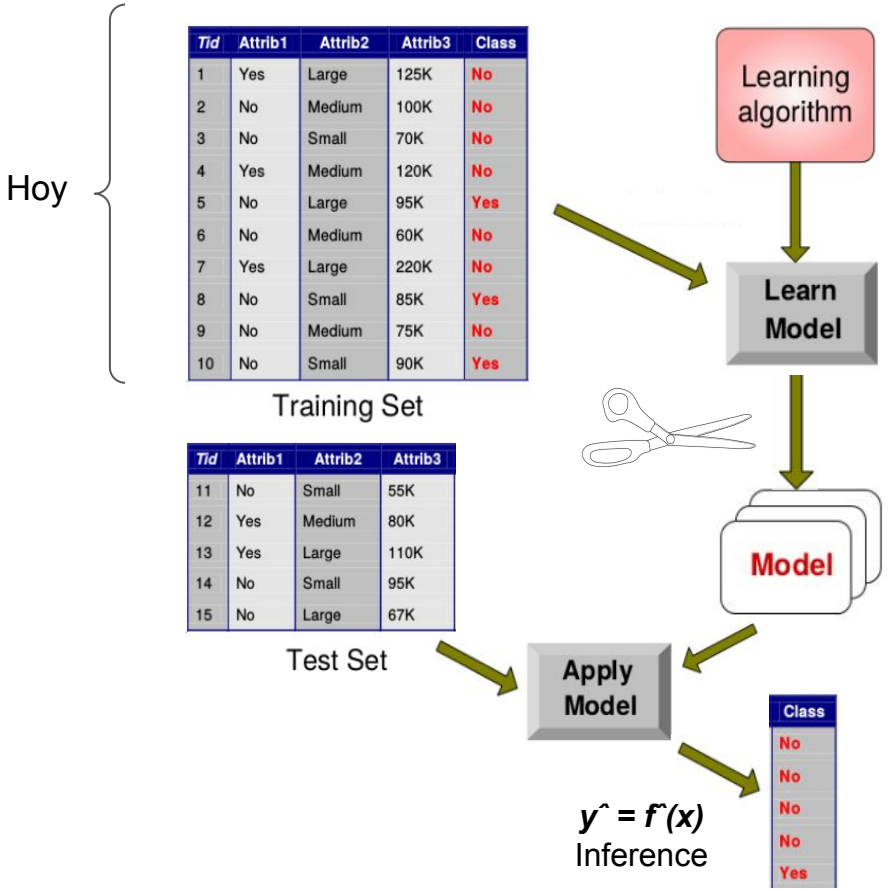
- Repaso de aprendizaje supervisado
- Ingeniería de atributos

Estructura de la clase

- Repaso de aprendizaje supervisado
- Ingeniería de atributos

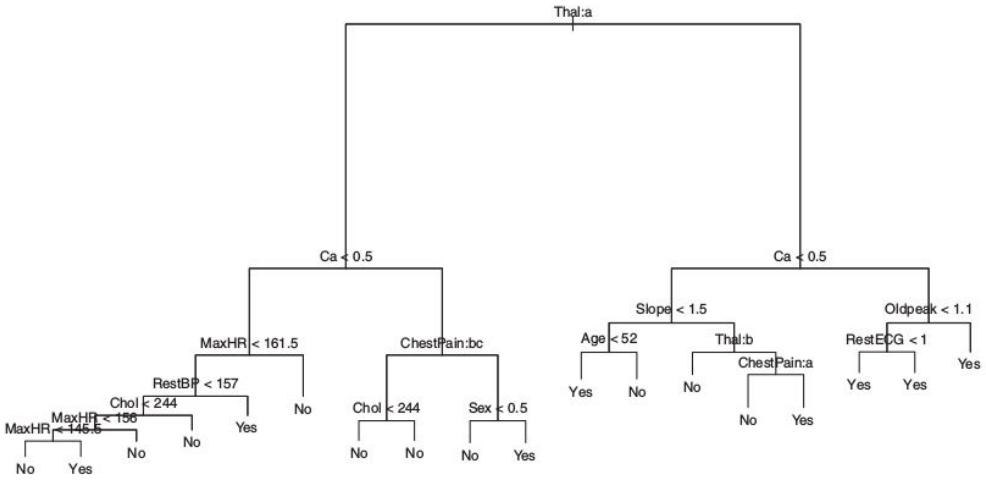
Repaso de aprendizaje supervisado

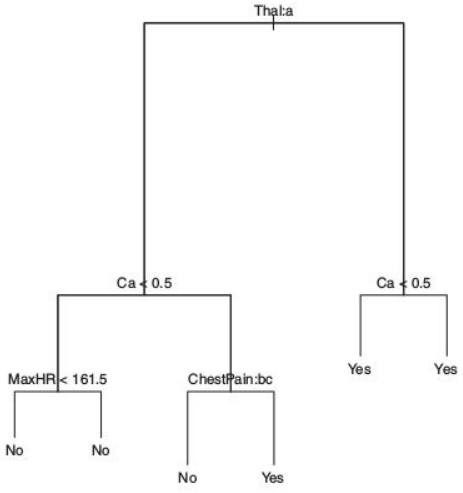
Esquema general de aprendizaje supervisado



Repaso de aprendizaje supervisado

Distintos modelos pueden tener mayor o menor flexibilidad. Incluso para un mismo algoritmo de aprendizaje esto ocurre





Repaso de aprendizaje supervisado

Animal	¿Da a luz?	¿Vuela?	¿Vive en el agua?	¿Tiene piernas?	¿Mamífero?
humano	sí	no	no	sí	mamífero
pitón	no	no	no	no	no-mamífero
salmón	no	no	sí	no	no-mamífero
ballena	sí	no	sí	no	mamífero
rana	no	no	a veces	sí	no-mamífero
dragón de komodo	no	no	no	sí	no-mamífero
murciélago	sí	sí	no	sí	mamífero
paloma	no	sí	no	sí	no-mamífero
gato	sí	no	no	sí	mamífero
tiburón leopardo	sí	no	sí	no	no-mamífero
tortuga	no	no	a veces	sí	no-mamífero
pingüino	no	no	a veces	sí	no-mamífero
puercoespín	sí	no	no	sí	mamífero
anguila	no	no	sí	no	no-mamífero
salamandra	no	no	a veces	sí	no-mamífero
monstruo de gila	no	no	no	sí	no-mamífero
ornitorrinco	no	no	no	sí	mamífero
búho	no	sí	no	sí	no-mamífero
delfín	sí	no	sí	no	mamífero
águila	no	sí	no	sí	no-mamífero

¿Da a luz?	¿Vuela?	¿Vive en el agua?	¿Tiene piernas?	¿Mamífero?
sí	no	sí	no	¿?

En clases se entrenó un modelo que predice una probabilidad igual a 0.8 de que un animal con estas características sea mamífero

¿Es una buena predicción?

Repaso de aprendizaje supervisado

Al momento de modelar, uno tiene muchas decisiones que tomar, por ej.:

- Qué datos tomar como inputs para entrenar modelo
- Qué preprocesamiento de los datos hacer
- Qué algoritmo de aprendizaje usar para predecir (*no free-lunch theorem*)
- Con qué valores de hiperparámetros entrenar el algoritmo de aprendizaje

Cada decisión seguramente impacte sobre la performance predictiva del modelo → Elegir un modelo que tenga buena performance en datos desconocidos es una tarea compleja

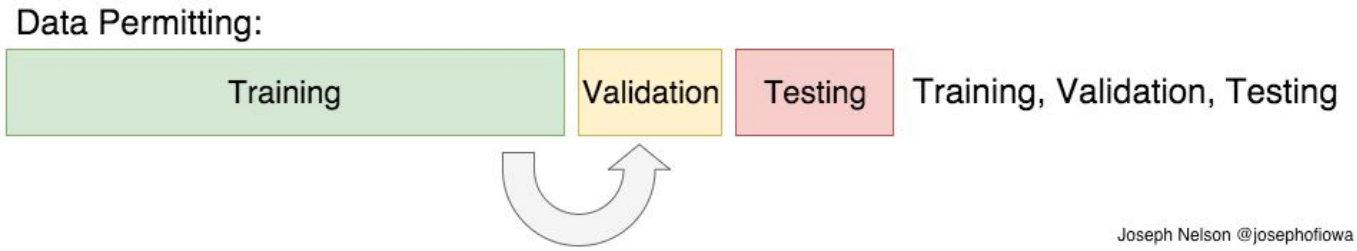
Repaso de aprendizaje supervisado

Estimaremos la performance que tendrá nuestro modelo en nuevas instancias de los datos

Puede servir para dos funciones:

- **Model selection**: consiste en estimar la performance de diferentes modelos para elegir el mejor
- **Model assessment**: después de haber elegido un modelo final, intentar estimar su performance en nuevos datos (*generalization error*)

Posible esquema (entre otros):

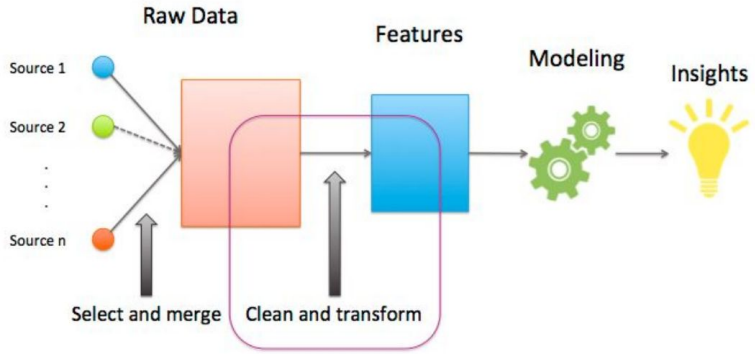


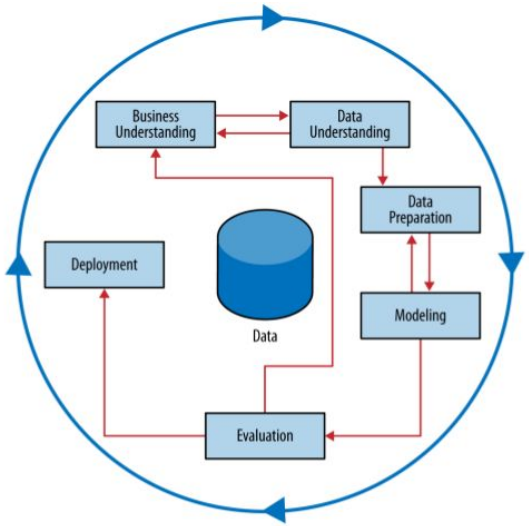
Estructura de la clase

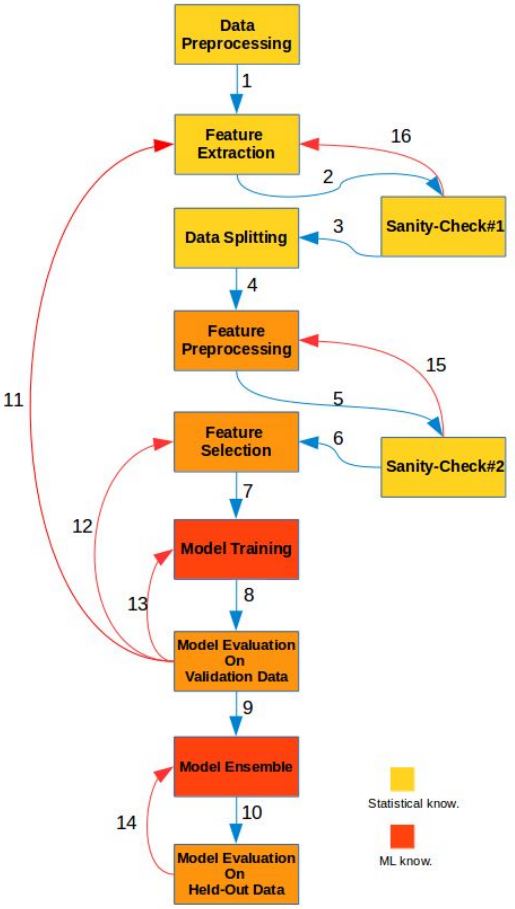
- Repaso de aprendizaje supervisado
- Ingeniería de atributos

Ingeniería de atributos

No importa de qué fuente tomemos un esquema del proceso aprendizaje, siempre se menciona como muy importante generar atributos para entrenar el modelo (*garbage in - garbage out*)







Ingeniería de atributos

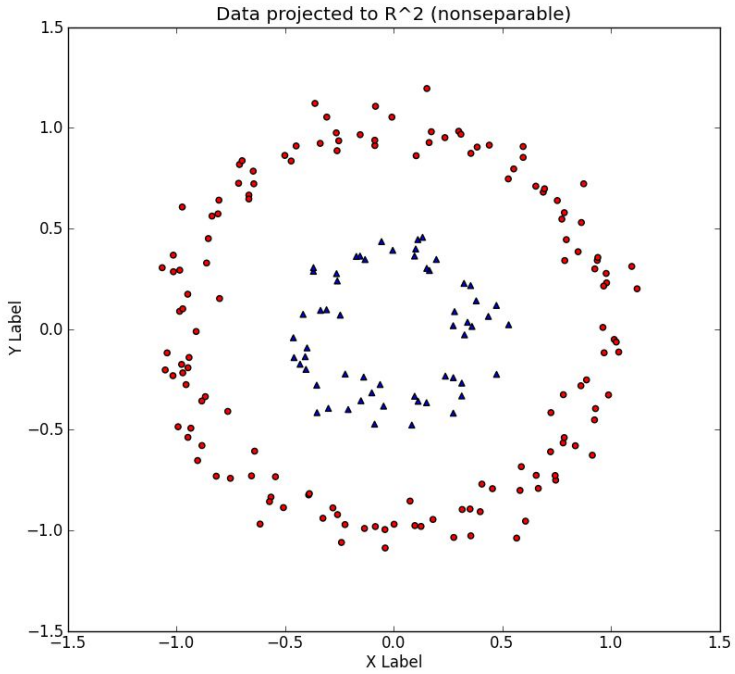
Lo que aprenda el algoritmo de aprendizaje dependerá de los atributos de los que disponga para aprender

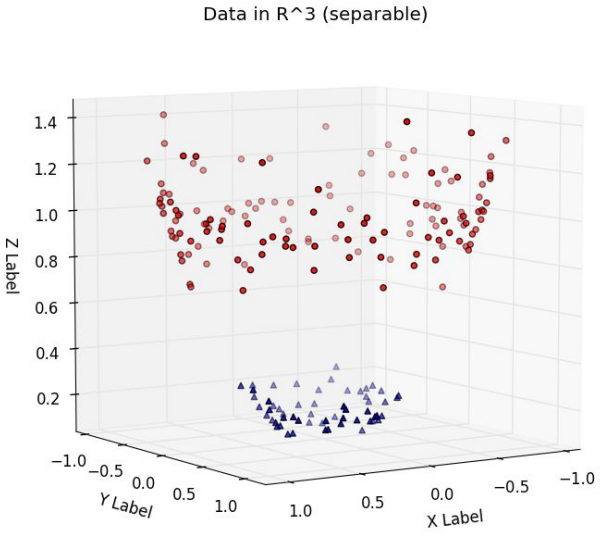
En soluciones ganadoras de competencias de aprendizaje automático se reporta que una porción importante del tiempo trabajado se centra en el análisis exploratorio de datos y en la "ingeniería de atributos":

- *Santander Product Recommendation*
- *Two Sigma Financial Modeling Code*
- *Rossmann Store Sales* (“*I spent 50% on feature engineering, 40% on feature selection plus model ensembling, and less than 10% on model selection and tuning*”)

Ingeniería de atributos

"*Good features should not only represent salient aspects of the data, but also conform to assumptions of the model. Hence transformations are often necessary*" (Zheng & Casari, Cap. 2)





$Z = X^2 + Y^2$

¿Dependerá también de la implementación del algoritmo?

Ingeniería de atributos - Aclaración 1

Veremos técnicas tradicionales de ingeniería de atributos. Dependiendo del dominio y del análisis exploratorio de los datos, pueden surgir nuevas ideas

IMPORTANTE: primero se debe disponer de un sistema de evaluación de modelos (holdout set, k-fold cross-validation... o lo que sea que replique a los datos "desconocidos"). Esto nos permitirá evaluar si las nuevas variables creadas/modificadas mejoran o no la performance de nuestro modelo

Seguiremos el siguiente orden de exposición:

1. Atributos numéricos
2. Valores faltantes
3. Atributos categóricos
4. Otras estrategias

Ingeniería de atributos - Aclaración 2

En el código de esta clase trabajaremos con tres clasificadores (los veremos en detalle en las siguientes clases). Por el momento, de los mismos, sólo mencionaremos lo que importa referido a procesamiento de datos

- Bayes Ingenuo (tiene mala fama, el nombre no ayuda):
  - Mejor para atributos categóricos (con continuos se suele asumir normalidad condicional a la clase)
  - No estima automáticamente linealidades e interacciones
- K-vecinos más cercanos (knn):
  - En su versión más simple utiliza la distancia euclídea entre el vector de atributos de la observación a predecir y cada una de las instancias de entrenamiento (este valor se ve afectado por la varianza de las variables)

$$\mathrm{d}(\mathbf{p}, \mathbf{q}) = \mathrm{d}(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

  - Mejor para atributos numéricos (con categóricos suele requerirse usar one-hot encoding)
- Regresión logística:
  - No estima automáticamente linealidades e interacciones

Ingeniería de atributos - Atributos numéricos

Log transformation

Muchas veces las variables siguen una distribución asimétrica positiva. Esto afecta a algunos modelos (knn, regresión lineal, regresión logística, etc.). Puede servir que el modelo trabaje con las variables asimétricas en escala logarítmica

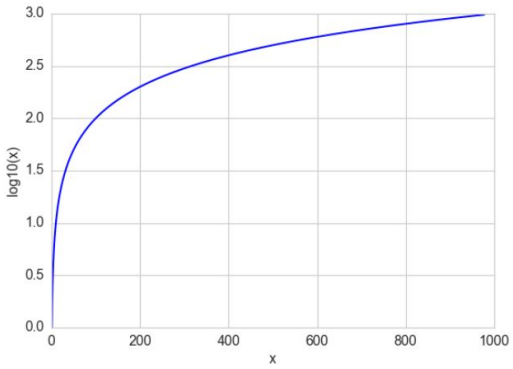
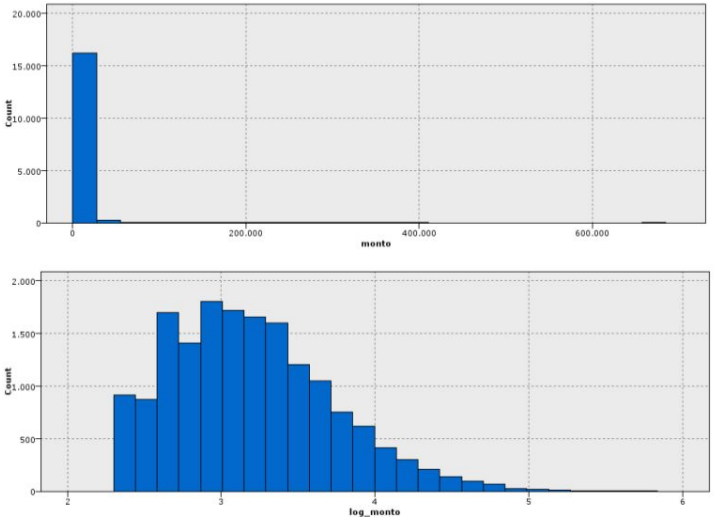


Figure 2-6. The log function compresses the high numeric range and expands the low range. Note how the horizontal x values from 100 to 1000 got compressed into just 2.0 to 3.0 in the vertical y range, while the tiny horizontal portion of x values less than 100 are mapped to the rest of the vertical range.



Si la variable original tiene valores no positivos una opción es transformarla con la siguiente expresión $x_{log} = log(x + 1 - min(x))$

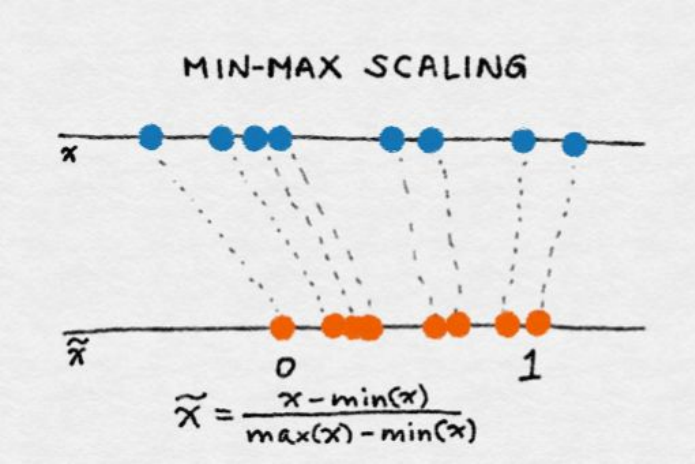
¿Debiera afectar esto en árboles de decisión?

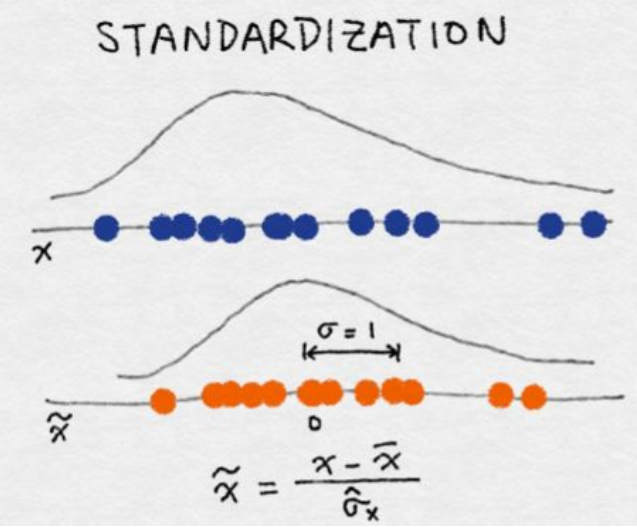
Ingeniería de atributos - Atributos numéricos

Feature Scaling or Normalization (variance scaling)

Muchos modelos son sensibles a la varianza de los atributos (por ej.: knn con distancia euclídea pondera más a atributos con alta varianza)

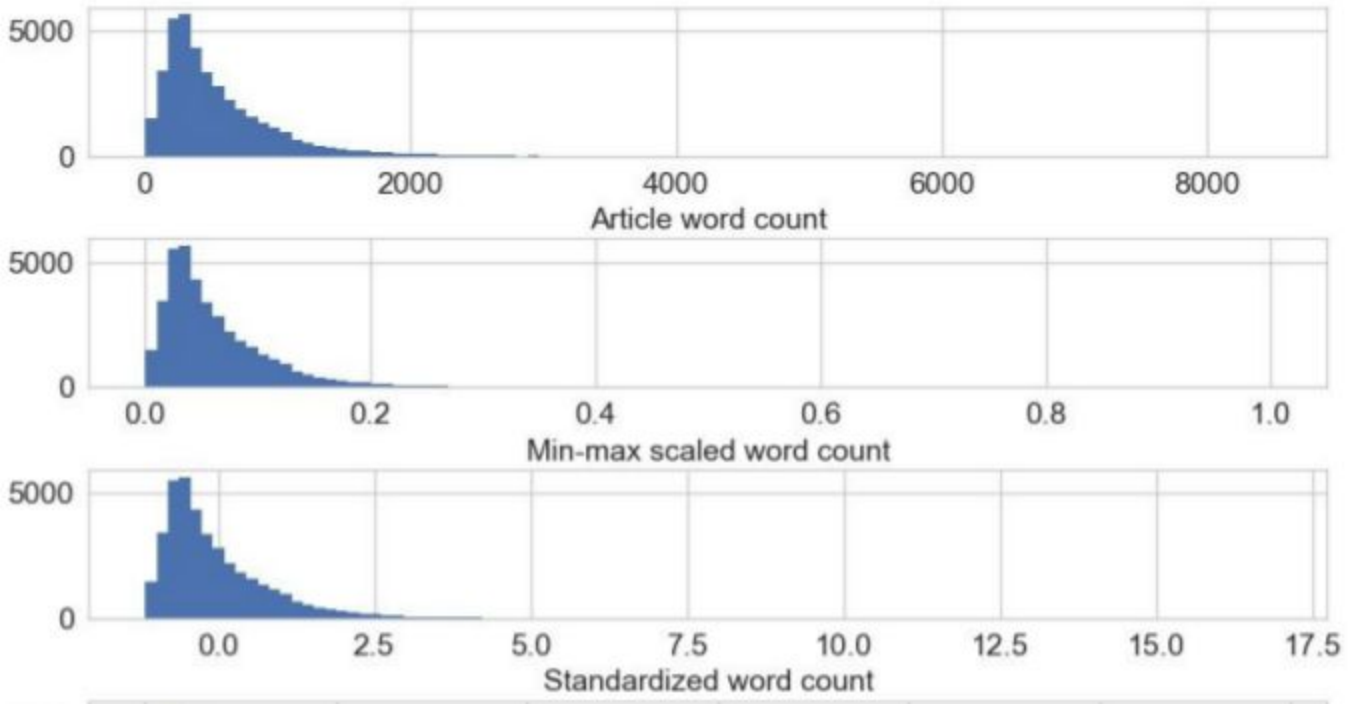
¿Debiera un modelo darle más importancia a, por ejemplo, la altura de una persona por estar expresada en centímetros en lugar de metros?





Ingeniería de atributos - Atributos numéricos

Noten que estas estrategias para escalar variables no modifican la "forma" de la distribución (sólo cambian los valores de los ejes)



Ingeniería de atributos - Atributos numéricos

Probemos el primer bloque de código

En el mismo se predice si una empresa entrará o no en bancarrota. Noten que la clase a predecir es desbalanceada

¿Parece buena decisión escalar las variables?

¿Mejora la performance sólo tomando logaritmos de las mismas?

¿Es aún mejor la performance si se combina tomar logaritmos con escalar variables?

Ingeniería de atributos - Atributos numéricos

Binarization

Hay situaciones donde podría sólo interesar si el valor de una variable numérica es mayor a una constante *c* (e.g., en conteos donde predominan los ceros, ver si algo es mayor a 0, o ver si un valor es mayor al valor mediano)

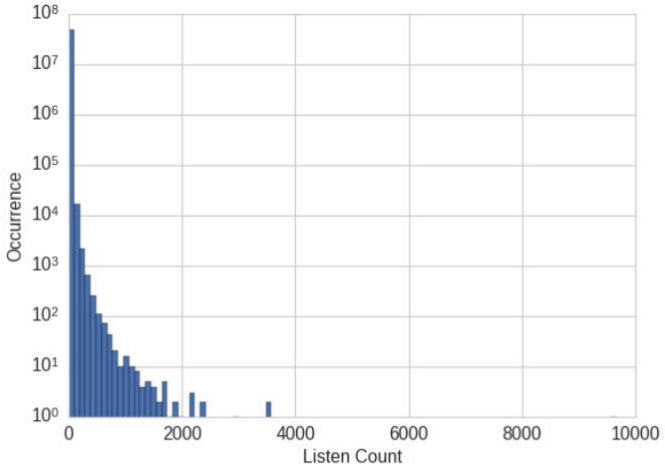


Figure 2-3. Histogram of listen counts in the user taste profile of the Million Song Dataset. Note that the y-axis is on a log scale.

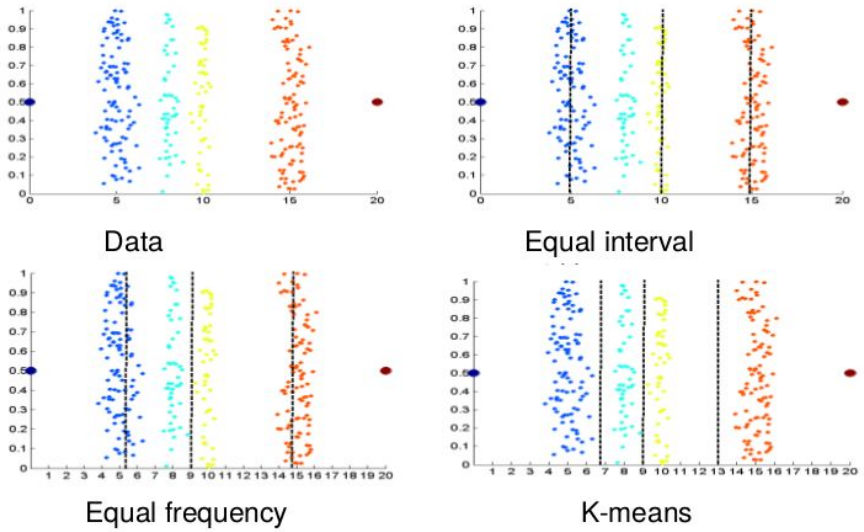
Se puede crear una nueva variable que valga 1 si la original es mayor a *c* y 0 en caso contrario (pudiendo la variable creada reemplazar a la original)

Ingeniería de atributos - Atributos numéricos

Quantization or binning

Consiste en agrupar las observaciones en bins (en donde los bins tienen un orden). Noten que las variables creadas son categóricas ordinales.

Existen múltiples criterios para elegir los cortes (uno totalmente válido es "a ojo")



Algunos algoritmos de aprendizaje supervisado internamente hacen esto (e.g., XGBoost con el hiperparámetros *max_bin*)

Ingeniería de atributos - Atributos numéricos

Probemos el segundo bloque de código

¿Tal cual vienen los datos, es buena la performance de Bayes ingenuo?

¿Mejora la performance discretizando las variables?

Ingeniería de atributos - Valores faltantes

Es muy común que un conjunto de datos tenga valores faltantes/desconocidos (¿1 + NA cuánto da?)

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

¿Por qué ocurren? ¿Son todos iguales?

Ingeniería de atributos - Valores faltantes

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

- **MCAR** (*missing completely at random*)

  P(missing) no depende de ninguna otra variable (observable o no observable).
  E.g.: Por error una persona no indicó su trabajo

- **MAR** (*missing at random*)

  P(missing) depende de valores de otras variables observables. E.g.: las personas con gender igual a “A” no reportan edad (o sólo algunas “A” lo hacen)

- **MNAR** (*missing not at random*)

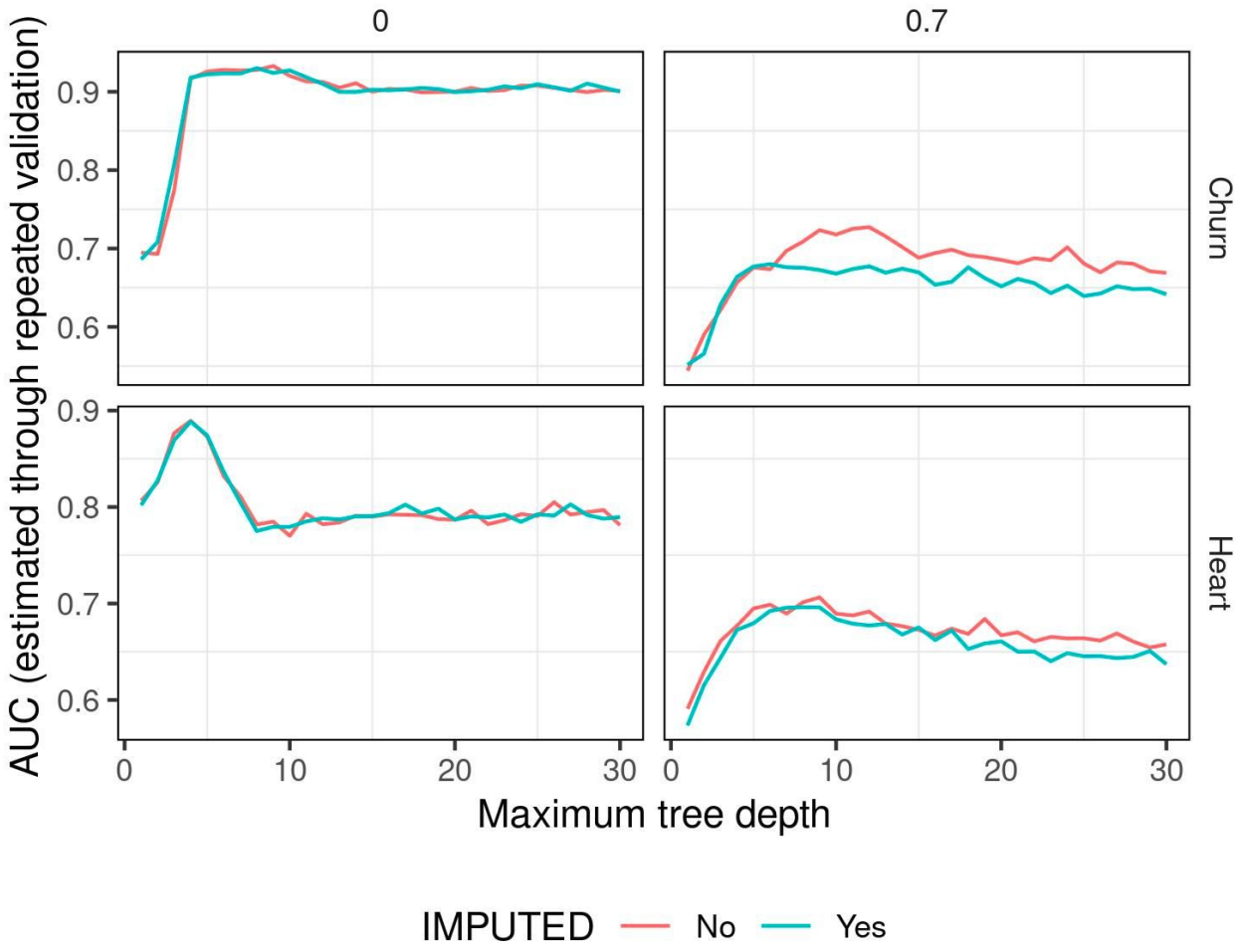
  P(missing) está relacionada con los valores de la misma variable. E.g.: las personas con altos salarios no reportan sus ingresos

Ingeniería de atributos - Valores faltantes

Posibles maneras de atacar el problema (en aprendizaje supervisado):

- Utilizar algoritmos que lidien con missings (no sklearn)
- Eliminar los valores faltantes (deletion)
  - ¿Eliminar las filas o las columnas?
  - ¿Está bien perder instancias de entrenamiento?
  - ¿Y qué hacer en producción?
- Para variables categóricas: considerar los missings como una nueva categoría (más en breve)
- Imputar (rellenar) valores faltantes
  - Estrategias generales:
    - Media / Mediana / Moda / Constante
    - Random Forest Imputer, KNN imputer, MICE, etc.
  - Estrategias para series temporales:
    - Last Observation Carried Forward (LOCF)
    - Next Observation Carried Backward (NOCB)
    - Interpolación (lineal, ponderada, splines, etc.)
  - Opción: al imputar, incorporar una columna adicional indicadora de missings

Ingeniería de atributos - Valores faltantes



Ingeniería de atributos - Atributos categóricos

Encoding Categorical Variables

No todos los algoritmos de aprendizaje supervisado (o sus implementaciones) trabajan con atributos categóricos. En estos casos uno debe transformar los atributos categóricos en numéricos intentando perder poca información

One-hot encoding:

Sample	Category
1	Human
2	Human
3	Penguin
4	Octopus
5	Alien
6	Octopus
7	Alien

Human	Penguin	Octopus	Alien
1	0	0	0
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
0	0	1	0
0	0	0	1

Problemático para variables con muchos niveles. Opciones:

- Uso de matrices esparsas/ralas (*sparse matrices*)
- Hashing-trick (más útil si se espera que haya nuevas categorías en validation, testing o producción)

¿Qué hacer con los missings?

Ingeniería de atributos - Atributos categóricos

Tipos de variables categóricas:

- **Nomimales**: las valores que puede tomar la variable son categorías mutuamente excluyentes que no tienen una jerarquía clara. Por ejemplo: nacionalidad de un cliente

- **Ordinales**: las valores que puede tomar la variable son categorías mutuamente excluyentes que sí tienen una jerarquía clara. Por ejemplo: nivel educativo de una persona

¿Se pierde información al hacer one-hot encoding sobre una variable categórica ordinal? ¿Es grave?

Posible estrategia: reemplazar los valores de las categorías de manera tal que la nueva variable numérica capture dicho orden

Ingeniería de atributos - Otros ejemplos variables derivadas

Otras alternativas a partir de variables categóricas:

- Conteos
- Promedios/proporciones de y_i según el valor de la variable categórica (*bin-counting*). IMPORTANTE: los promedios deben ser calculados sobre training

n	usuario	y
1	A	0
2	B	1
3	B	1
4	A	0
5	B	0
6	A	0
7	A	1
8	C	0
9	A	0
10	B	1

usuario_conteo	usuario_bc
5	0.2
4	0.75
4	0.75
5	0.2
4	0.75
5	0.2
5	0.2
1	0
5	0.2
4	0.75

Ingeniería de atributos - Interacciones (*feature crossing*)

Algunos modelos no capturan automáticamente interacciones y no linealidades (por ejemplo, los modelos lineales)

Estrategia: generar una nueva matriz de atributos que contenga combinaciones polinómicas de los atributos originales, con un grado igual o menor al grado especificado

Ejemplo con tres predictores (X_1, X_2 y X_3) y grado 2:

$$(X_1, X_2, X_3) \rightarrow (X_1, X_2, X_3, X_1{*}X_2, X_1{*}X_3, X_2{*}X_3, X_1^2, X_2^2, X_3^2)$$

Por ejemplo, para clasificación, en el nuevo espacio generado quizás sea más simple separar linealmente las clase (¿peligro de overfitting?)

Ingeniería de atributos - OHE e Interacciones

Probemos el tercer bloque de código

¿Mejora la performance de regresión logística al incorporar interacciones?

Ingeniería de atributos - Otros ejemplos variables derivadas

A partir de un único timestamp:

- Generar nuevos atributos de, por ejemplo, año, mes, día, hora, minuto, día de la semana, semana del año, día desde el 1970-01-01 (tendencia), etc.

ID	Datetime	Count	year	month	day	dayofweek_num	dayofweek_name
0	2012-08-25 00:00:00	8	2012	8	25	5	Saturday
1	2012-08-25 01:00:00	2	2012	8	25	5	Saturday
2	2012-08-25 02:00:00	6	2012	8	25	5	Saturday
3	2012-08-25 03:00:00	2	2012	8	25	5	Saturday
4	2012-08-25 04:00:00	2	2012	8	25	5	Saturday

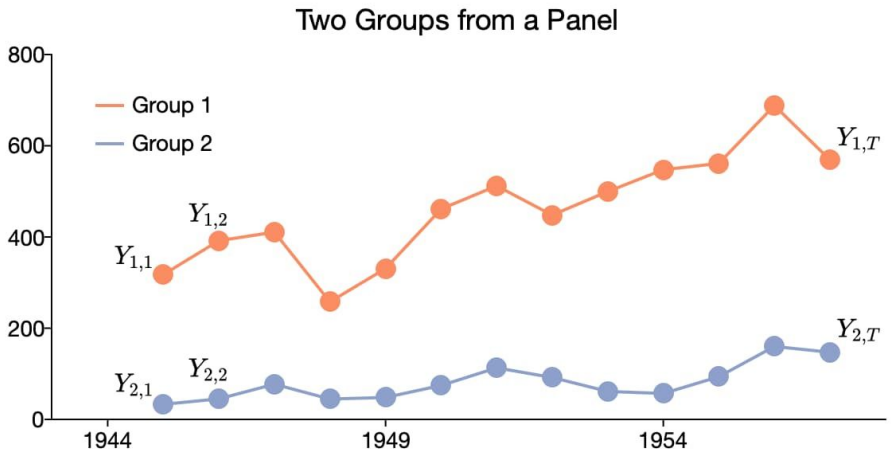
A partir de más de un timestamp:

- Calcular diferencias entre los timestamps (e.g.: la antigüedad de un cliente se calcula como "*timestamp de hoy - timestamp de su alta*")

Ingeniería de atributos - Otros ejemplos variables derivadas

Datos con estructura de panel: (ejemplo, evolución de los saldos de los clientes de un banco mes a mes)

Group	Time Period	Notation
1	1	Y_{11}
1	2	Y_{12}
1	T	Y_{1T}
⋮	⋮	⋮
N	1	Y_{N1}
N	2	Y_{N2}
N	T	Y_{NT}



Algunas estrategias:

- Considerar medias móviles, máximos móviles, mínimos móviles, desvíos estándar móviles, etc. para cada grupo
- Considerar tendencias (i.e., ¿para cada grupo, el atributo está subiendo, bajando o estable?)

Bibliografía

- Zheng, & Casari, “Feature Engineering for Machine Learning”, Capítulos 2 y 5

- Huyen, “Designing Machine Learning Systems: An Iterative Process for Production-Ready Applications”, Capítulo 5 (salvo lo referido a embeddings)

Manejo de pipelines en sklearn (opcional)

- Müller & Guido, "Introduction to machine learning with Python: a guide for data scientists”, Capítulo 6