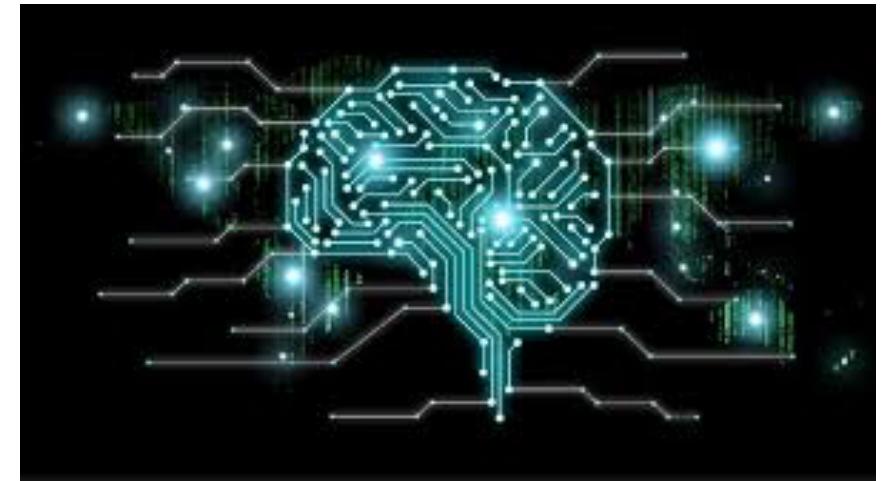


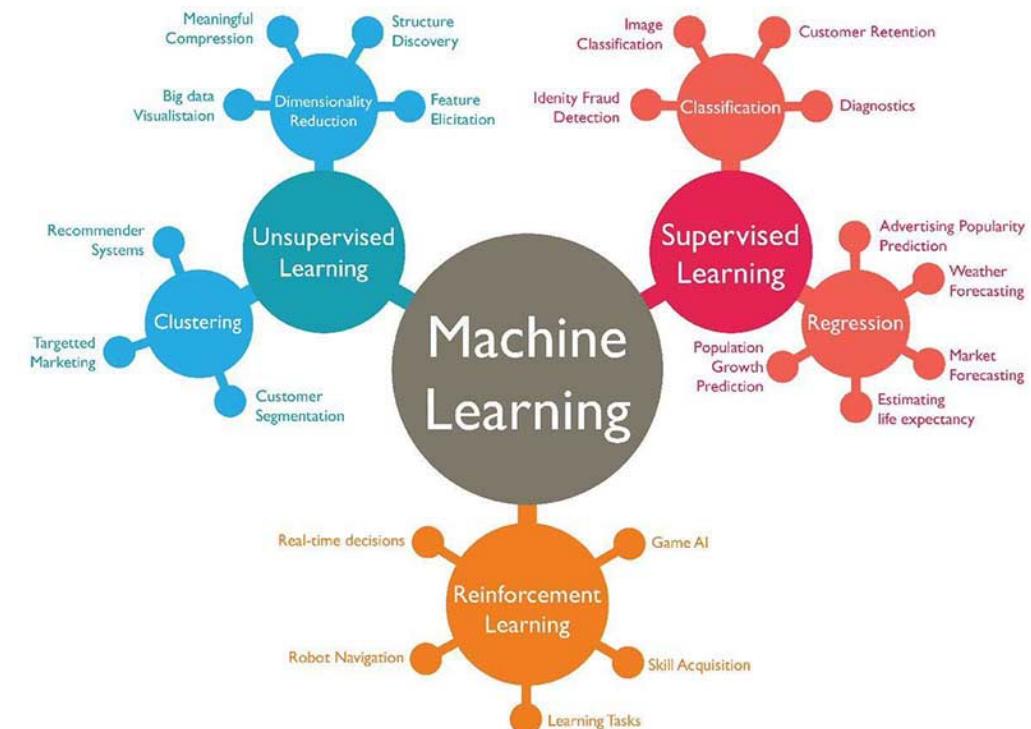
¿Qué es el Machine Learning (ML)?

- Disciplina de la IA centrada en la extracción de conocimiento y patrones a partir de una serie de observaciones (datos).
- Extracción de conocimiento sin necesidad de programar los sistemas para que lo hagan
 - Permite obtener valor de los datos sin realizar grandes esfuerzos de programación
- Requieren cantidades importantes de datos.



Técnicas de Machine Learning

- Supervisado
 - Datos etiquetados
 - Modelos predictivos de clasificación y regresión
- No supervisado
 - Datos no esquematizados
 - Técnicas para extraer estructuras y patrones de los datos.
- Aprendizaje por refuerzo
 - Cuando el sistema es retroalimentado sobre lo buena o mala que son algunas decisiones.



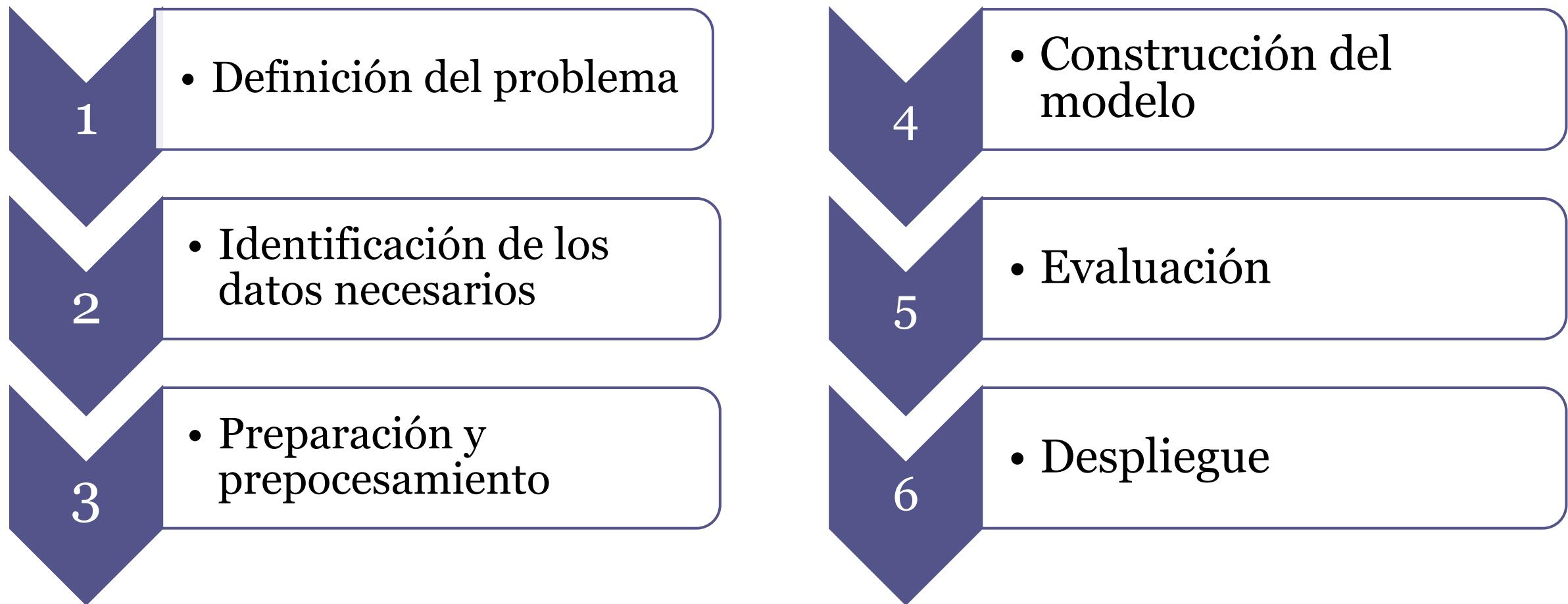
<https://www.techleer.com/articles/203-machine-learning-algorithm-backbone-of-emerging-technologies/>

¿Qué podemos hacer?

- Gestión de clientes:
 - Predecir perdidas de clientes (customer churn)
 - Customer Scoring
 - Predicción de demanda
 - Predecir decisiones de los clientes
- Industria
 - Mantenimiento predictivo
 - Predicción de consumo energético
 - Predicción de roturas de stock



Fases de un proyecto de Minería de Datos



1. Definición del problema

- Identificación los objetivos organizacionales:
 - ¿Qué problema estamos intentando resolver?
 - Adquisición de nuevos clientes
 - Retención de clientes
 - Reducción de los costes de mantenimiento y operacionales, ...
 - ¿Qué medidas vamos a utilizar para determinar si se han cumplido los objetivos
- Identificación los objetivos de modelo predictivo
 - De objetivos organizacionales a objetivos de minería de datos
 - Primera aproximación a qué datos son necesarios
 - ¿Cuáles van a ser las entradas y las salidas del problema?
 - Definir la metodología a utilizar
 - Definir objetivos de desarrollo alcanzables

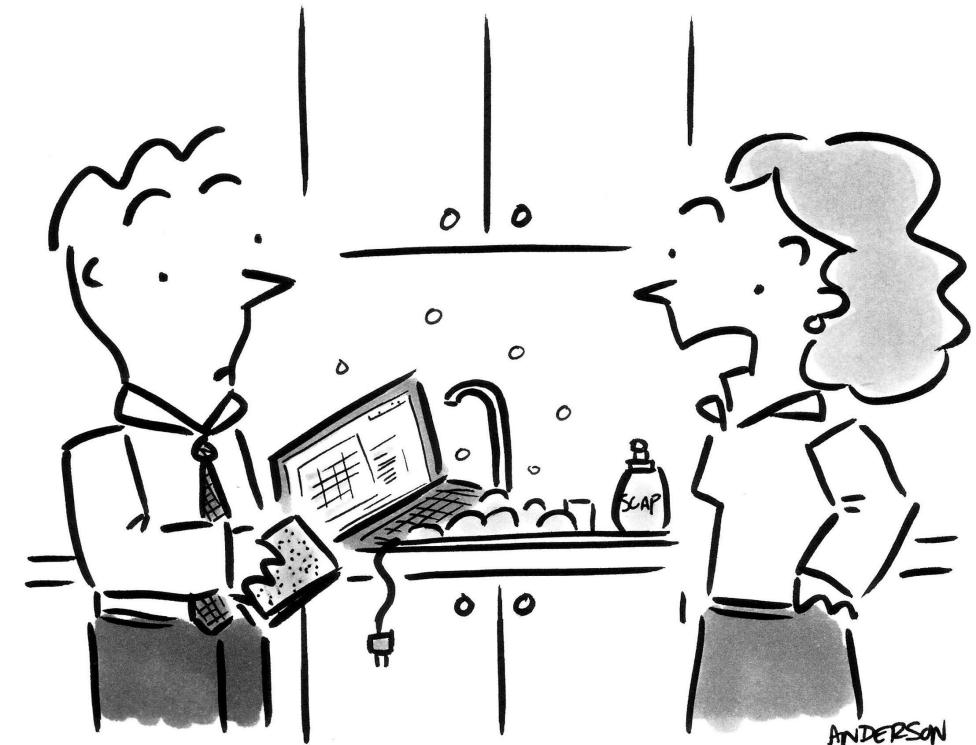
2. Identificación de los datos necesarios

- Recoger, describir y explorar los datos
 - Identificar las fuentes de datos
 - Analizar y comprender los metadatos
 - Analizar la calidad de los datos
 - Bad data vs. Good data
 - Herramientas de análisis datos, estadística, técnicas de visualización
 - Interacción entre los ingenieros de datos y los expertos de la organización



3. Preparación y preprocesamiento de datos

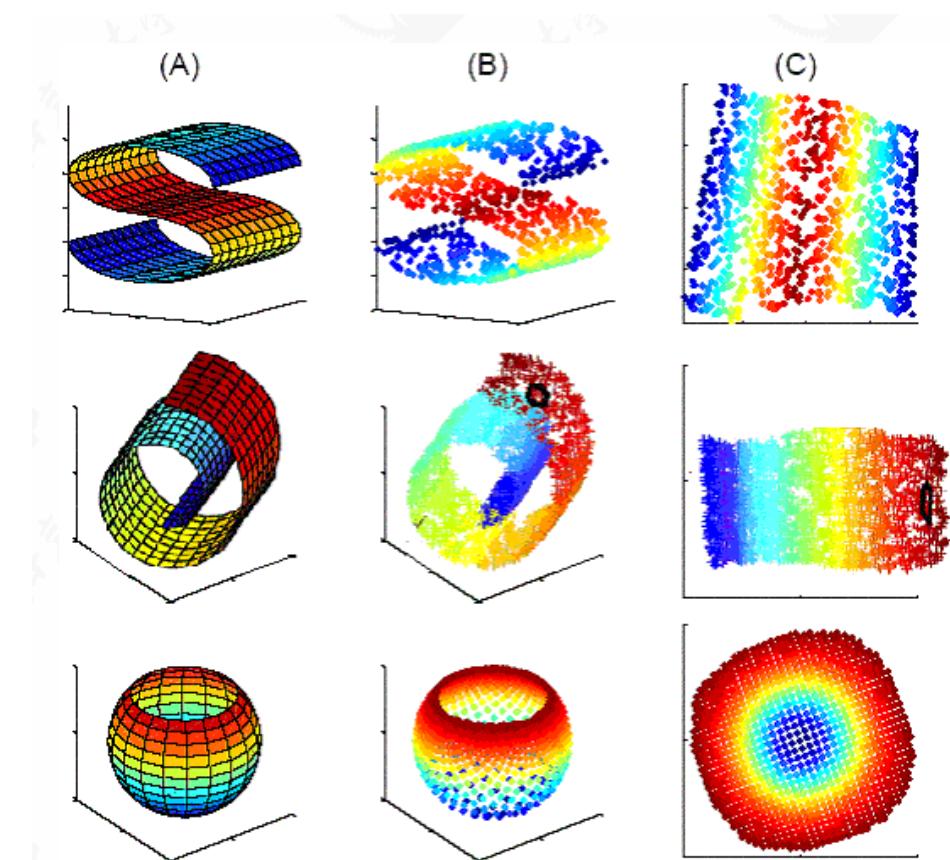
- Construcción del modelo de datos para la fase de desarrollo de los modelos predictivos
 - Integración de datos
 - Limpieza de los datos
 - Transformación del formato de los datos
 - Creación de atributos derivados
 - ...
 - ETL



"This is not what I meant when I said 'we need better data cleansing!'"

4. Proceso de construcción del modelo

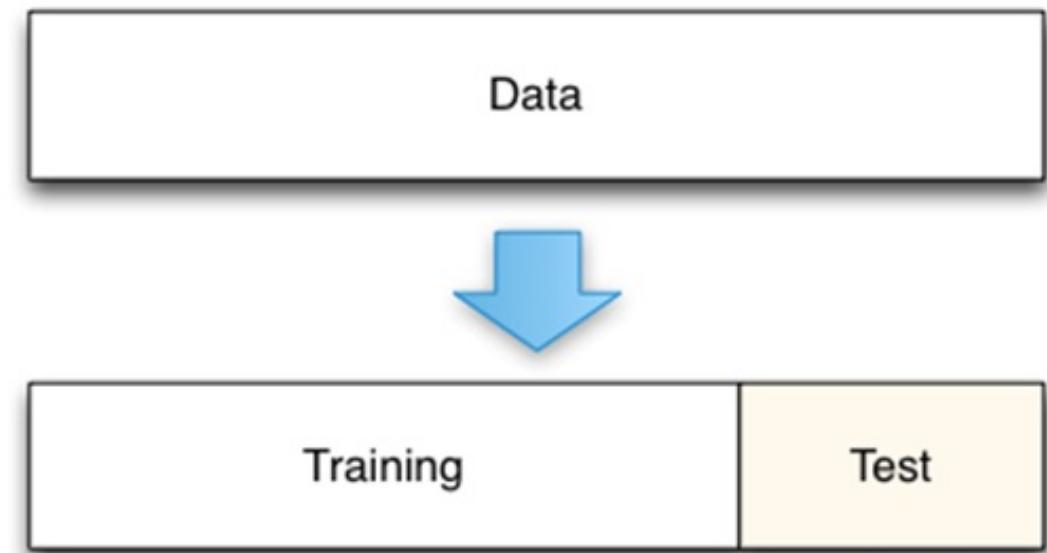
- 4.1 Preprocesado de datos
 - Análisis del estado del arte
 - Feature Engineering/Extraction
 - Adaptar los datos a las técnicas utilizadas
 - Variables dummy.
 - Discretización.
 - Eliminación de ruido.
 - Cambios de escala....
 - Valores ausentes
- 4.2 Selección de variables.
 - Filtros
 - Wrappers
 - Reducción de la dimensionalidad



<http://jntsai.blogspot.com/2015/04/ammai-nonlinear-dimensionality.html>

4. Proceso de construcción del modelo

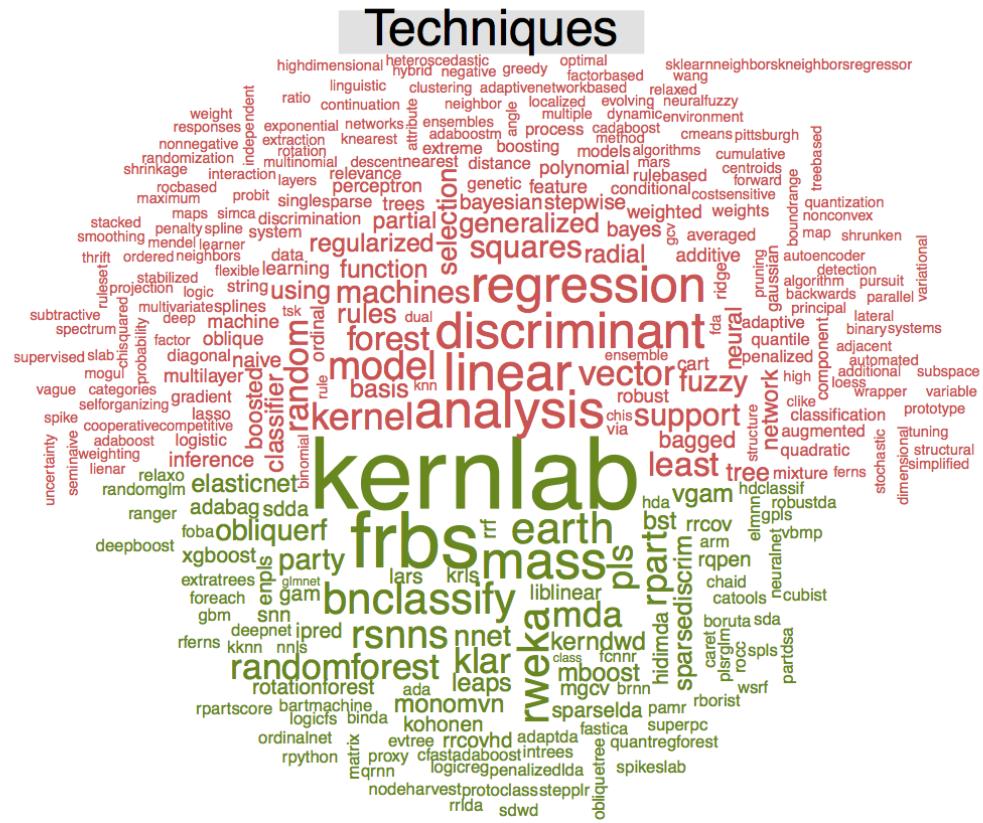
- 4.3 Creación de los conjuntos de training y test:
 - Necesario para evaluar la capacidad de generalización del modelo
 - Nos permitirá detectar el overfitting.
- 4.4 Selección de la técnica de muestreo:
 - Hold-out
 - Cross-validation
 - Bootstrap
 - A tener en cuenta
 - Estratificación
 - Repetición



<http://scott.fortmann-roe.com/docs/docs/MeasuringError/holdout.png>

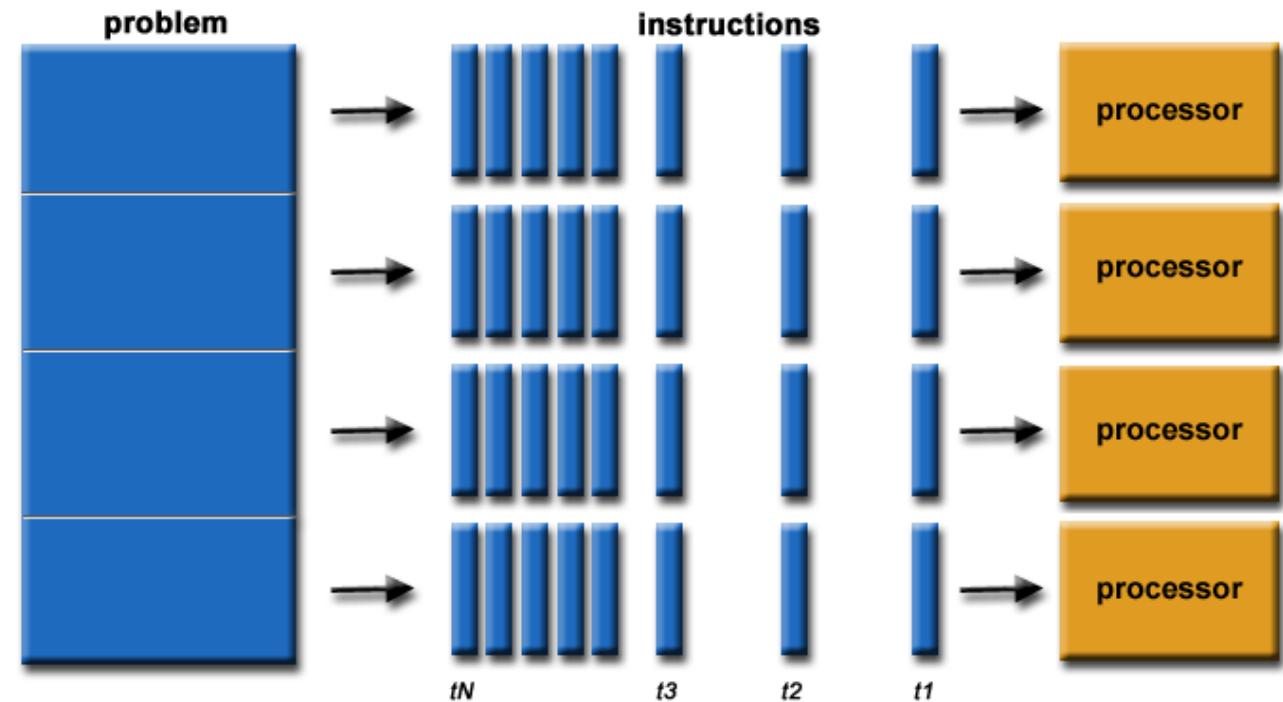
4. Proceso de construcción del modelo

- 4.5 Selección de las diferentes técnicas y aplicarlas
 - Análisis del estado del arte
 - ¿Necesitamos modelos interpretables?
 - Analizar y comprender los parámetros de los modelos
 - ¿Qué características de los modelos afectan a los datos?
 - 4.7 Búsqueda de hiperparámetros
 - ¿Qué parámetros hay que determinar?



4. Proceso de construcción del modelo

- 4.6 Training
 - Reproducibilidad
 - Paralelización
 - Alto nivel a nivel de experimentos
 - Bajo nivel a nivel de técnicas



<https://www.kdnuggets.com/2016/11/parallelism-machine-learning-gpu-cuda-threading.html>

4. Proceso de construcción del modelo

- Ejemplo de selección de características:
 - Se prueban con conjuntos de tamaño 3 a 19
- Se garantiza la reproducibilidad
- Wrapper
 - Random Forest para evaluar
 - Validación cruzada con 5 folds
- Esto hace $16^* 5 = 80$ modelos!!!

```
subsets <- c(3:19)

set.seed(123)
seeds <- vector(mode = "list", length = 6)
for(i in 1:5) seeds[[i]] <- sample.int(1000, length(subsets) + 1)
seeds[[6]] <- sample.int(1000, 1)

registerDoParallel(cl <- makeCluster(8))

ctrl.treebag <- rfeControl(functions=rfFuncs, method = "cv", number = 5,
                           seeds = seeds, returnResamp="final",
                           verbose = TRUE, allowParallel = TRUE)

rf.tb <- rfe(Churn~, data=customerTrain, sizes=subsets,
            rfeControl=ctrl.treebag)
```

4. Proceso de construcción del modelo

- Ejemplo de selección de training:
- Se garantiza la reproducibilidad
- Máquina de soporte de vectores lineal
 - Búsqueda de hiperparámetros
 - Grid de tamaño 9
 - Hold-out stratificado con 10 repeticiones
- Esto hace $10^*9 = 90$ modelos!!!

```
set.seed(123)
```

```
seeds <- vector(mode = "list", length = 11)
for(i in 1:10) seeds[[i]]<- sample.int(n=1000, 54) #for the last model
seeds[[11]]<-sample.int(1000, 1)
```

```
registerDoParallel(cl <- makeCluster(8))
```

```
fitcontrol <- trainControl(method = "LGOCV", p=.75, number=10,
                           seeds = seeds, returnResamp = "final",
                           verbose=FALSE, allowParallel = TRUE)
```

```
set.seed(342)
```

```
svmGrid <- expand.grid(.C=c(0.25,0.5,1,2,4,8,16,32,64))
```

```
svmFit <- train(Churn~, data=customerTrain.sel,
                 method="svmLinear", tuneGrid=svmGrid,
                 trControl=fitcontrol)
```

5. Evaluación

- Evaluar si los modelos generados cumplen la expectativas.
 - **Primera fase:** sólo expertos en minería de datos
 - Seleccionar los criterios de evaluación
 - Medidas de rendimiento
 - Puede implicar volver a la anterior fase
 - Seleccionar los mejores/mejor modelo

Performance metrics

For each class (or for two class problems):

Precision / PPV	$tp / (tp + fp)$	$\text{green} / (\text{green} + \text{red})$
Recall / Sensitivity	$tp / (tp + fn)$	$\text{green} / (\text{green} + \text{orange})$
Specificity	$tn / (tn + fp)$	$\text{blue} / (\text{blue} + \text{red})$
Accuracy	$(tp+tn) / (tp + fp + fn + tn)$	$(\text{green} + \text{blue}) / (\text{green} + \text{orange} + \text{red} + \text{blue})$
F1-score	$2 * \text{prec} * \text{sens} / (\text{prec} + \text{sens})$	

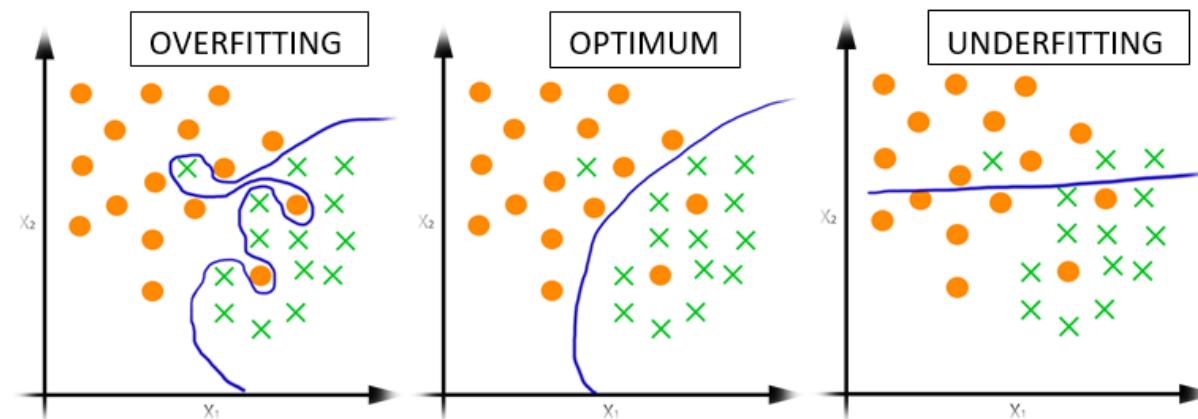
Basic elements for each class:

- true positives
- false positives
- false negatives
- true negatives

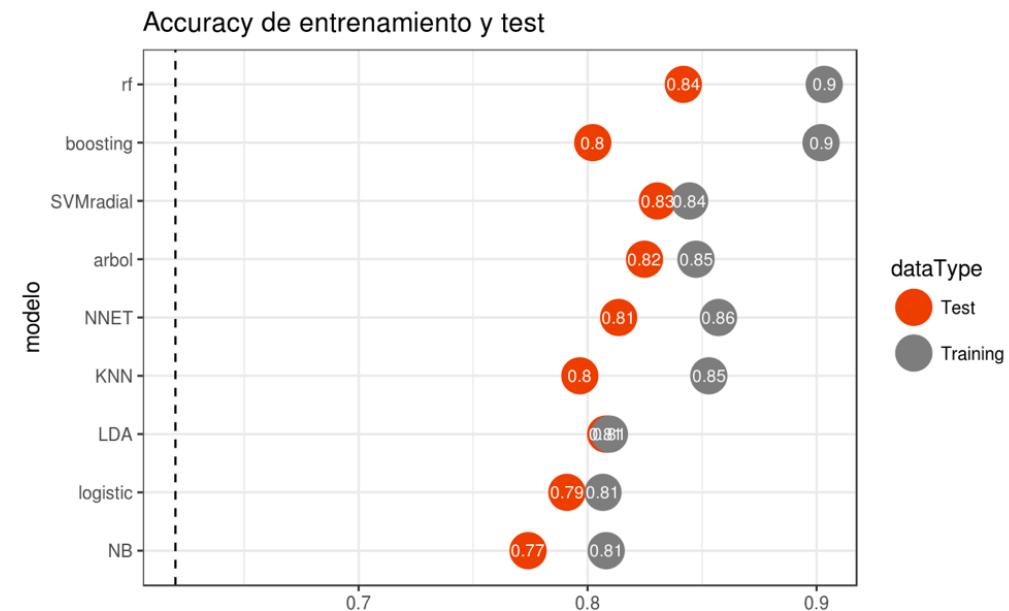


5. Evaluación

- Detectar el overfitting.
 - Evitar que los modelos “memorizan” los datos.
 - Los resultados obtenidos en training suelen ser optimistas
 - Hay que evaluar en test (datos no utilizados en training).

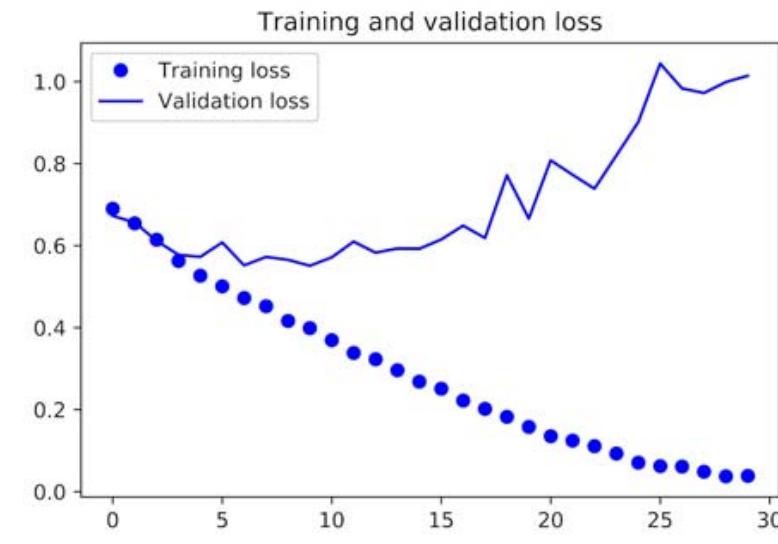
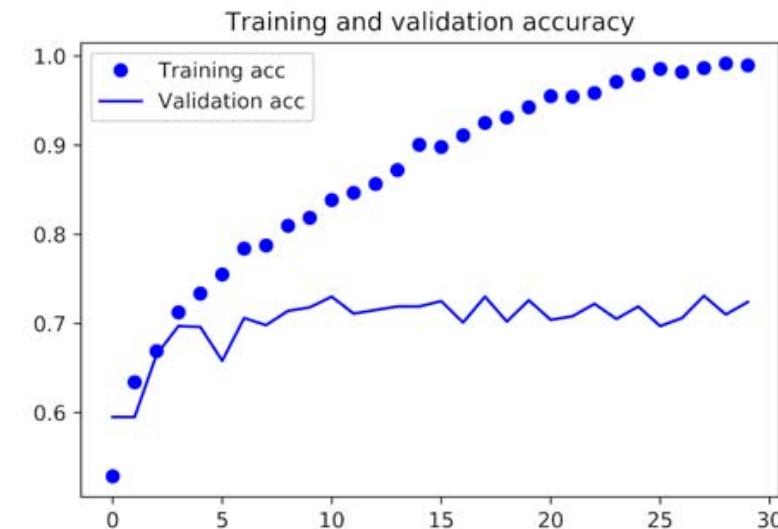


<https://medium.com/@srjoglekar246/overfitting-and-human-behavior-5186df1e7d19>



5. Evaluación

- Overfitting en Deep Learning.
 - Como evitarlo
 - Reduciendo las iteraciones (epoch)
 - Dropout
 - Regularización



5. Evaluación

- **Segunda fase:** analizar el modelo en el contexto organizacional
 - ¿El modelo satisface los objetivos organizacionales?
 - ¿Hemos tenido en cuenta todos los aspectos organizacionales?



6. Despliegue

- Integrar el modelo en la infraestructura tecnológica
 - Debe estar planificado con antelación
 - Facilitado por las herramientas utilizadas
 - Análisis de rendimiento
- Mantenimiento
 - ¿Cuándo se debe reentrenar el modelo?
 - Utilizar los indicadores establecidos en la primera fase
 - Considerar la aportación de nuevos datos

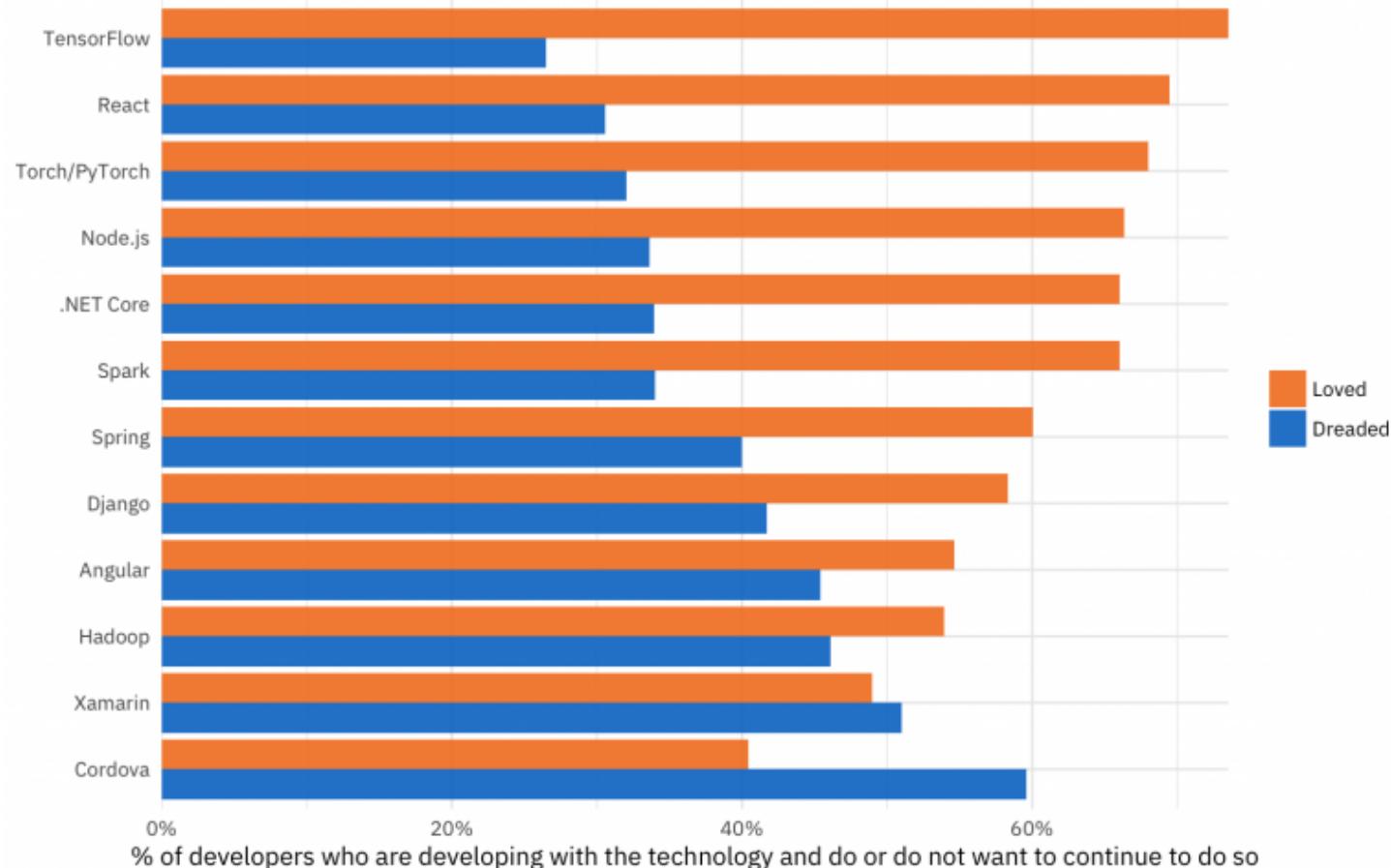


6. Herramientas



Most loved and dreaded frameworks

On the 2018 Stack Overflow Developer Survey, TensorFlow is the most loved framework



6. Herramientas

KDnuggets 2018 Data Science, Machine Learning Software Poll:
Top Tools Associations



6. Herramientas

Lenguajes	Plataformas	Big Data	Open Source tools
<ul style="list-style-type: none">• Python<ul style="list-style-type: none">• Scikit-learn• Tensorflow• Keras,• Theano,• PyTorch• R<ul style="list-style-type: none">• Caret• Keras• C++,C#,Java, ..	<ul style="list-style-type: none">• Google Cloud Machine Learning Platform• Microsoft Machine Learning Studio• AWS machine learning• Watson Machine Learning• Oracle Machine Learning	<ul style="list-style-type: none">• Apache Spark MLlib• Hadoop• Node.js	<ul style="list-style-type: none">• Accord.NET• KNIME• Weka• Orange• Shogun

Conclusiones

- Necesidad de equipos multidisciplinares
- Es importante la formación
 - No se pueden utilizar las herramientas de forma automática
 - Hay que saber bajo qué condiciones se pueden aplicar cada modelo
 - Hay que saber interpretar los resultados.
- Gracias a las herramientas la aplicación del machine learning se ha convertido en un proceso de ingeniería.
 - Importancia de la selección de la herramienta y lenguaje
- A la hora de desplegar infraestructuras tecnológicas tener en cuenta las posibles y futuras aplicaciones.