



Procesamiento de Señales II

Ciencia de Datos II

Análisis y Curación - Reproducibilidad

Dr. José Ramón Iglesias

DSP-ASIC BUILDER GROUP

Director Semillero TRIAC

Ingeniería Electronica

Universidad Popular del Cesar

Reproducibilidad

El proceso que seguimos
depende del tipo de producto
de datos que se busca obtener

Análisis de un dataset

El producto final es la descripción del fenómeno:

- Censos poblacionales
- Cálculo de índices de desarrollo
- Análisis de segmentos de mercado

Proceso:

1. Recolección de datos
2. Análisis y exploración
3. Extracción de conclusiones

Producto final:

1. Descripción y entendimiento del fenómeno

Investigación de tecnologías

El producto final es un prototipo o metodología novedosa

- Mejorar el estado-del-arte en traducción automática
- Comparación de modelos para recomendar asignaciones de subsidios

Proceso:

1. Recolección de datos
2. Análisis y exploración
3. Pre-procesamiento del conjunto de datos
4. Experimentación para encontrar el mejor modelo
5. Extracción de conclusiones

Producto final:

1. Descripción y entendimiento del fenómeno y los modelos
2. Modelo entrenado

Servicios basados en datos

El producto es un servicio que provee respuestas

- Recomendador de canciones
- Traductor automático

Proceso:

1. Entrenamiento:
 - a. Recolección de datos históricos
 - b. Análisis y exploración
 - c. Pre-procesamiento del conjunto de datos
 - d. Experimentación para encontrar el mejor modelo
2. Produccionalización:
 - a. Recolección de NUEVOS datos para predecir
 - b. Pre-procesamiento del conjunto de datos
 - c. Aplicación del modelo

Producto final:

1. Sistema de predicción

Crisis de reproducibilidad en la ciencia

The booming field of artificial intelligence (AI) is grappling with a replication crisis, much like the ones that have afflicted psychology, medicine, and other fields over the past decade.

<https://science.sciencemag.org/content/359/6377/725>

(Facebook) When combined with the unavailability of code and models, the result is that the approach is very difficult, if not impossible, to reproduce study, improve upon, and extend.

<https://arxiv.org/abs/1902.04522>

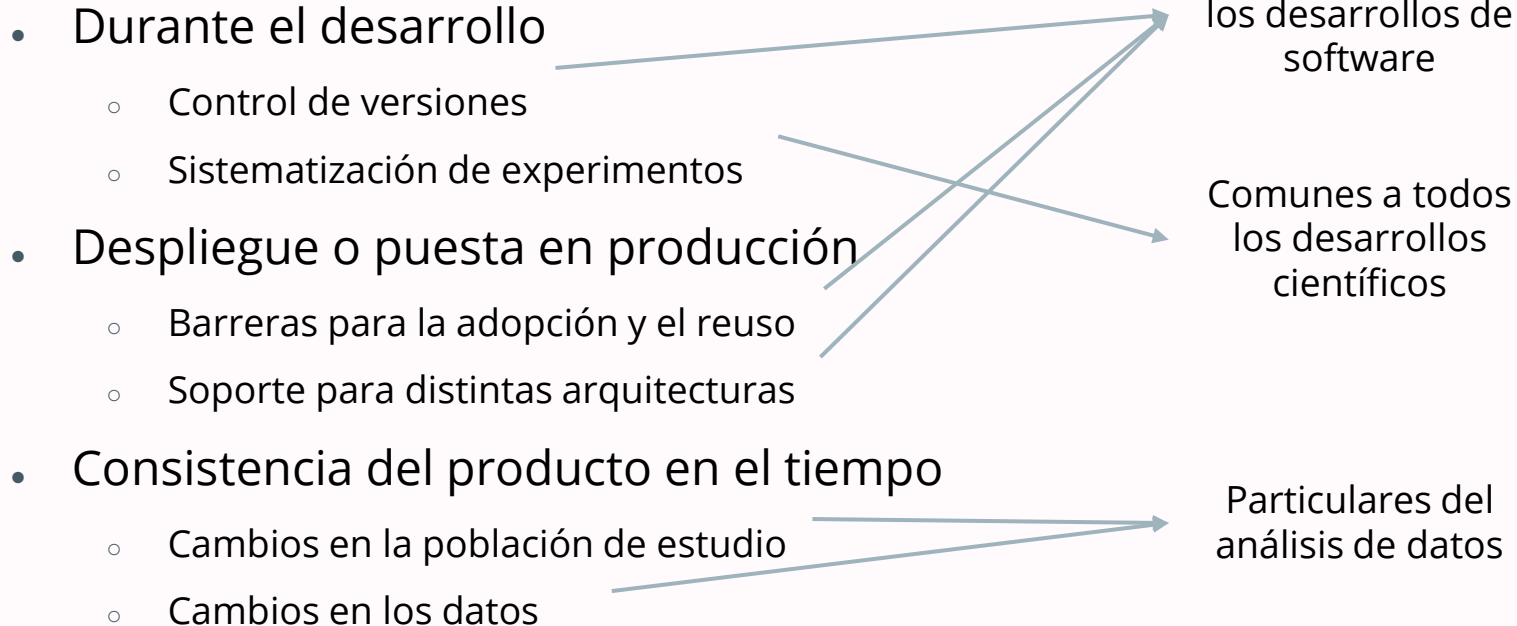
(Google) ML systems have a special capacity for incurring technical debt, because they have all of the maintenance problems of traditional code plus an additional set of ML-specific issues.

<https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf>

Even the original author sometimes couldn't train the same model and get similar results!

<https://petewarden.com/2018/03/19/the-machine-learning-reproducibility-crisis/>

Aspectos de la reproducibilidad



Recomendaciones para lograr mejores
resultados

Durante todo el proceso

Metodología de la investigación

- Documentar, documentar, documentar.... y actualizar la documentación vieja.
- Disponibilizar los datos originales. Nunca sobre-escribirlos
- Tener un documento *Journal* donde escriben informalmente qué conclusiones sacaron ese día.

Metodología de la investigación

- Llevar un registro formal de los resultados experimentales
[Ejemplo real]

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Q
1			Dev Results											Dataset	Embedding
2	Log	Name	Accuracy	Ac-std	Precision	P-std	Recall	R-std	F1-Score	F1-std	Time/Feat	Activation	Attention	Config	Char embe
3	echr_none_none zx	18-11-14-14-49	0.661	0.070	0.655	0.086	0.661	0.070	0.652	0.082	None	None	No	Explore	cnn
4		18-11-14-16-39	0.611	0.071	0.630	0.061	0.611	0.071	0.611	0.065	None	None	No	Explore	None
5		18-11-14-18-15	0.635	0.050	0.702	0.060	0.635	0.050	0.641	0.048	None	None	No	Explore	cnn
6		18-11-14-20-17	0.620	0.056	0.669	0.082	0.620	0.056	0.628	0.058	None	None	No	Explore	lstm
7		18-11-14-22-02	0.616	0.036	0.637	0.036	0.616	0.036	0.617	0.028	None	None	No	Explore	None
8	echr_time_sigmoid	18-11-14-22-22	0.633	0.079	0.657	0.079	0.633	0.079	0.636	0.077	Word	Sigmoid	Yes	Explore	lstm
9		18-11-15-00-38	0.640	0.085	0.697	0.089	0.640	0.085	0.648	0.090	Word	Sigmoid	Yes	Explore	cnn
10		18-11-15-02-17	0.636	0.030	0.684	0.038	0.636	0.030	0.644	0.030	Word	Sigmoid	Yes	Explore	cnn
11		18-11-15-04-16	0.653	0.065	0.683	0.072	0.653	0.065	0.660	0.071	Word	Sigmoid	Yes	Explore	None
12		18-11-15-06-01	0.645	0.071	0.688	0.075	0.645	0.071	0.648	0.072	Word	Sigmoid	Yes	Explore	lstm
13		18-11-15-09-54	0.642	0.074	0.689	0.071	0.642	0.074	0.653	0.073	Word	None	Yes	Explore	None
14	echr_time_sigmoid	18-11-15-10-16	0.651	0.057	0.687	0.053	0.651	0.057	0.651	0.061	Word	Sigmoid	Yes	Definitive	lstm

Durante el desarrollo

Trabajo sobre notebooks

Ventajas

- Rapidez de configuración
- Rapidez de desarrollo
- Interactividad para agilizar la exploración
- Permite agregar documentación al análisis

Desventajas

- Difícil de mantener un control de versiones
- Variables es estados potencialmente inconsistentes
- No se pueden ejecutar programáticamente (por ejemplo, con un script)

Estructuremos mejor la base de código

- Separar la exploración del pre-procesamiento de los datos
- **No** incluir archivos con datos en el repositorio
- Automatizar la mayor cantidad de tareas posibles. Por ejemplo, entrenamiento de modelos
- Extraer los bloques de código que estén repetidos. Por ejemplo: chequeos y transformaciones durante la lectura de datos

Ejemplo de repositorio

project_name

- INSTALL.md
- **models**
 - best_knn.py
- **notebooks**
 - Prices exploration.ipynb
 - Coordinates exploration.ipynb
 - Experiment Results.ipynb
- README.md
- **preprocess**
 - add_airbnb_data.py
 - impute_missing_years.py
- run_preprocess.py
- run_experiment_best_knn.py
- **tests**
 - test_best_knn.py

Configuraciones

- Utilizar control de versiones y repositorios.
- Guardar registro de las versiones utilizadas para cada librería.
 - Lo más fácil: usar entornos virtuales como conda
 - Lo más avanzado: usar empaquetadores como Docker
- Utilizar documentos README.md para guardar instrucciones de ejecución e instalación *junto con el código*

Objetivo: que cualquier
persona pueda instalar y
recrear sus resultados dentro
de 1 año

Durante el despliegue
(deploy)

Evaluar los requerimientos del producto

Buscar la herramienta adecuada (que seguro ya existe). Ejemplos:

- Código que acompaña un paper => disponibilizar a través de un repositorio
- Librería para clasificación de imágenes => empaquetar usando Docker para que pueda ejecutarse en cualquier sistema.
- Procesamiento de 10TB de imágenes => usar Spark sobre un sistema de archivos distribuido

¿Existe la sobre-ingeniería de procesos?

Esfuerzo que lleva aprender y aplicar una herramienta específica

vs

Beneficio que aporta la herramienta

Material adicional

- [Tutorial de Docker](#) en castellano
- Guía [Essential Skills for reproducible Research Computing](#)
- <https://awesome.re/> Listas de software abiertos activos y recomendados por la comunidad. Ordenados por equipos o por lenguaje: