

Actualizado 01/09/2022


## Pruebas realizadas con fichero 16k noticias

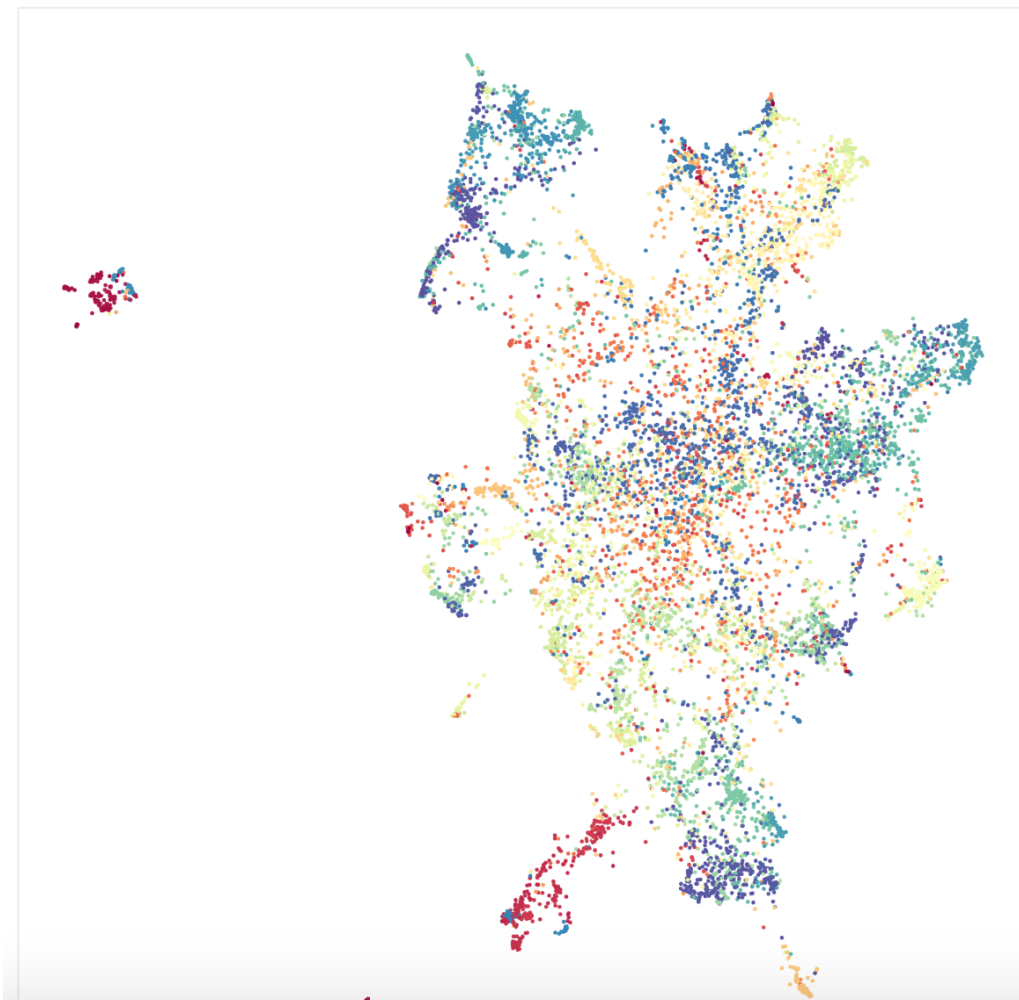
- **Top2vec** (transformer + U-map + DB-Scan).

(umap: <https://pair-code.github.io/understanding-umap/>)

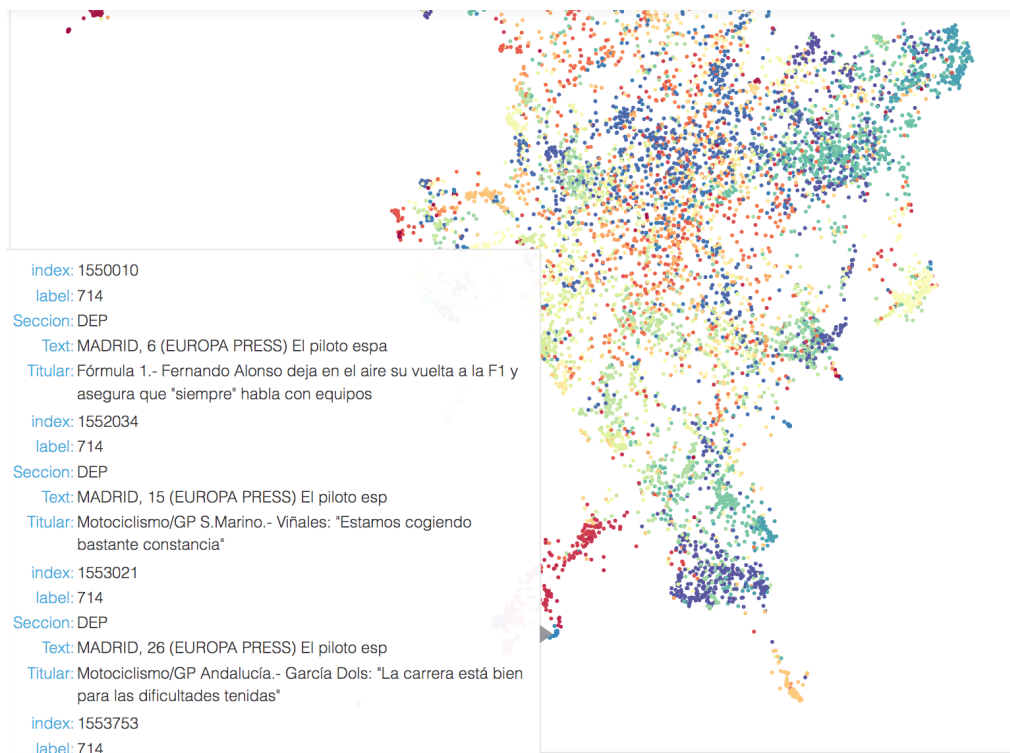
Mapa global interactivo obtenido:

```
len data = 1559390  
len topic_vectors = 791  
document_vectors generados.
```

 BokehJS 2.4.3 successfully loaded.



Zooms:



**Conclusiones:** clasificación y visualización tiene bastante sentido. Las clases mejor segmentadas son aquellas que son más frecuentes en el corpus, como por ejemplo la de deportes. Notebook Pedro, pedro/top2vec\_adaptado.ipynb

## **LDA (Latent Dirichlet Analysis)**

- LDA (Pedro): hay que seleccionar bien el número de tópicos (15). Conclusiones: salen tópicos con sentido.
- NER de spacy: Pedro cogió noticias concretas y chequeó entidades. Conclusiones: a ojo, sacaba las entidades bien. También hizo tf-idf enfatizando las entidades extraídas por NER y luego sacaba LDA. Conclusiones: saca prácticamente lo mismo que con el LDA sin NER.

## **Pruebas iniciales clasificación fake-no fake**

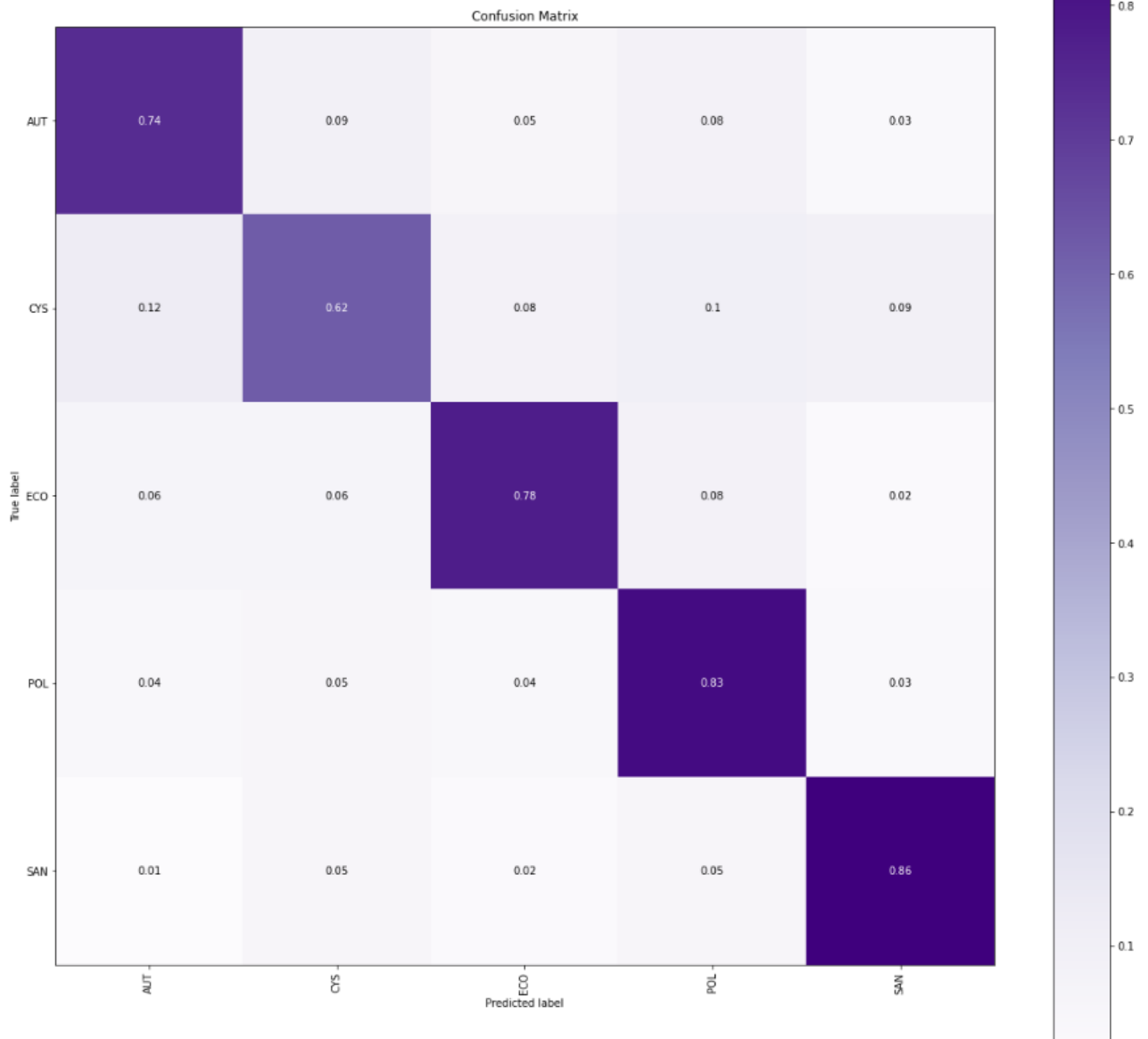
- Detección de fakes con clasificador fake-no fake preentrenado en 1000 noticias descargado por Fran. Da resultados ok (salen marcadas como fake noticias que luego EP confirma que lo son, concretamente las noticias absurdas; también sacaba como fake noticias que no eran realmente noticias sino listados de eventos o anuncios).
- Detección de anomalías (Fran) con autoencoder sobre vectores de documentos extraídos por top2vec: resultados malos (pero no está seguro de si lo hizo bien).

## **Pruebas realizadas con fichero 1M noticias**

- Top2vec adaptado (transformer + U-map + DB-Scan). La adaptación consistía en ir cargando datos en bloques y grabar resultados parciales a fichero ya que había problemas de memoria con todo el dataset. Conclusión: se obtenían clases que tenían sentido, pero los resultados no eran similares a los obtenidos con fichero 16k ya que ahora las clases están muy desbalanceadas. EP mandó noticias adicionales para aliviar desbalanceado pero no eran suficientes. Trabajo que podemos hacer: undersampling de las categorías.

## **Otras pruebas**

- Explicabilidad de modelos de clasificación de secciones usando shap. Librería utilizada: shap. Conclusiones: la herramienta de visualización da resultados intuitivos. Resultados (Notebook en máquina virtual mv1-europapress:
- Primera prueba con todas las clases ('AEX', 'AUT', 'CUL', 'CYS', 'DEP', 'ECO', 'EDU', 'INV', 'MOT', 'OCI', 'POL', 'SAN', 'TRI'), se observaron FN entre las clases EDU y (CYS, o SAN), entre MOT y ECO y entre OCI y CYS,.
- La siguiente prueba ha sido incluir única y exclusivamente las clases que formen un conjunto igual o superior al 10% del total de los datos (AUT, CYS, POL, ECO, SAN). Obteniendo unos mejores resultados en la clasificación y obteniendo resultados realmente buenos de verdaderos positivos y verdaderos negativos.



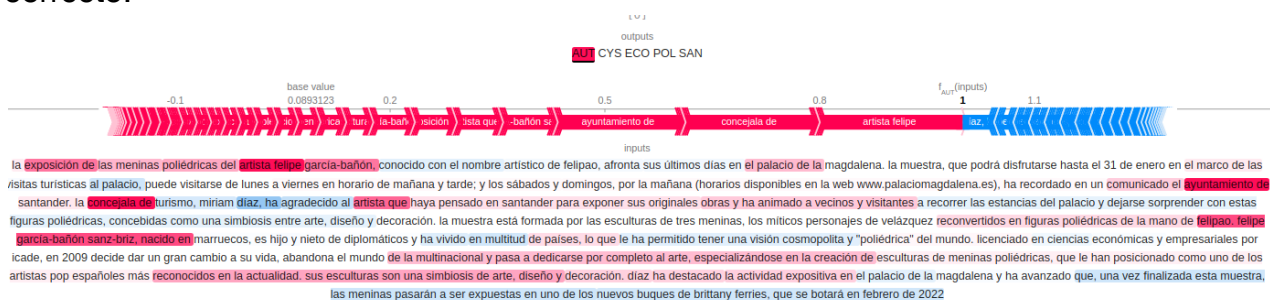
- /fran/supervised\_topic\_classification-pruebas\_5\_clases.ipynb VERSIÓN 0):

-

-

AUT:

correcto:

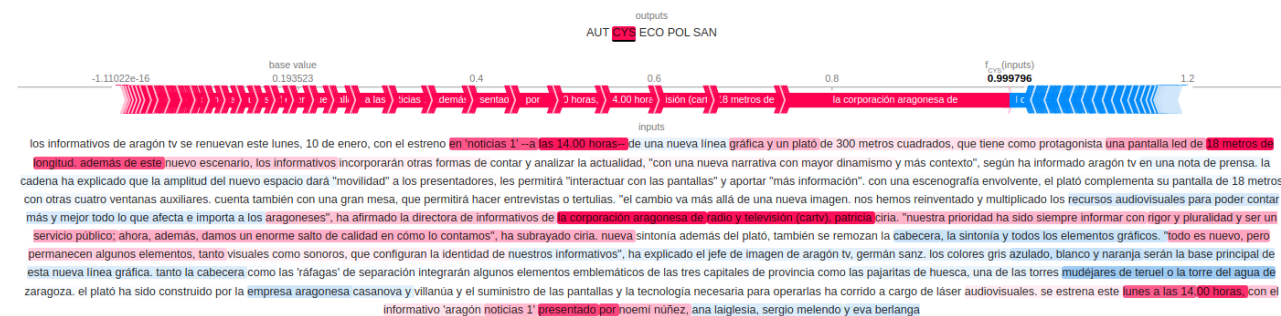


incorrecto:

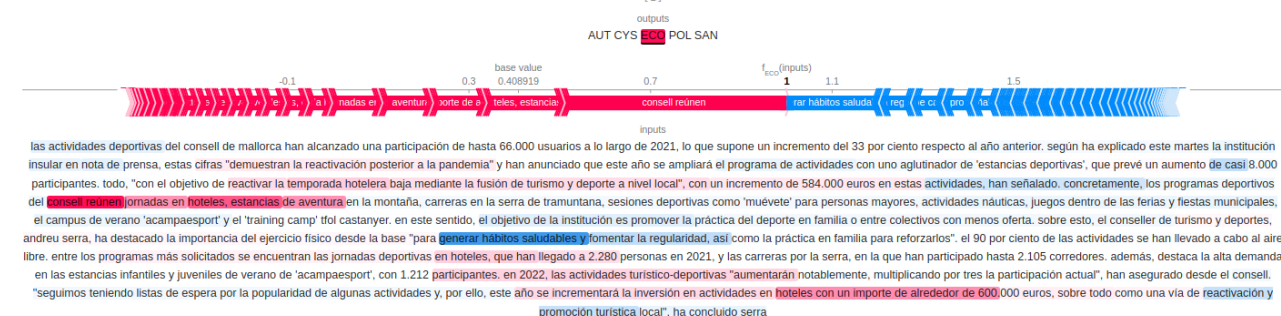


CYS:

correcto:

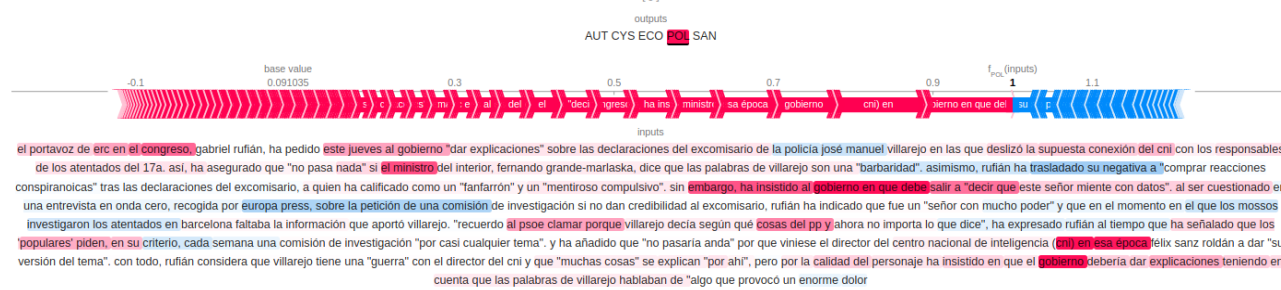


incorrecto:



POL:

correcto:



incorrecto:



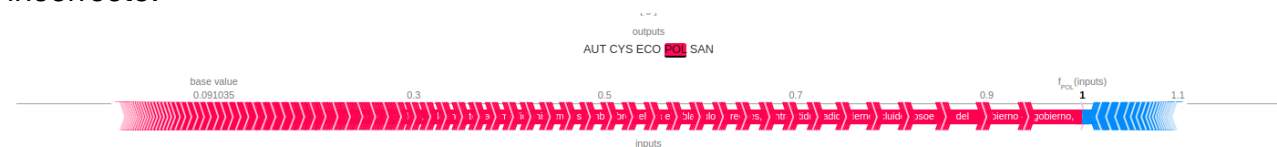
la gestora de fondos blackrock cerró el ejercicio 2021 con un beneficio neto atribuido de 5.901 millones de dólares (5.150 millones de euros), lo que equivale a un incremento del 19,6% en comparación con el año pasado, según ha informado este miércoles la compañía; al cierre del último ejercicio, los activos bajo gestión de blackrock ascendían a 10.010 billones de dólares (8.73 billones de euros). una cifra que supone un incremento anual del 15,4%, gracias al atractivo de los fondos de inversión cotizados (etfs), en todo el año pasado, la entidad contabilizó entradas netas por importe de 540.000 millones de dólares (471.313 millones de euros), lo que implica un crecimiento orgánico de los activos del 6%, impulsado por el crecimiento récord de los flujos hacia etfs y estrategias activas. la cifra de negocio de blackrock en el conjunto de 2021 alcanzó los 19.374 millones de dólares (16.910 millones de euros), un 19,5% más, incluyendo un crecimiento del 14% entre octubre y diciembre, hasta los 5.106 millones de dólares (4.456 millones de euros), de este modo, en el cuarto trimestre, la gestora obtuvo un beneficio neto atribuido de 1.643 millones de dólares (1.434 millones de euros), un 6,1% por encima del resultado contabilizado un año antes. "blackrock generó el crecimiento orgánico más fuerte de nuestra historia, incluso cuando nuestros activos bajo administración alcanzaron nuevos máximos", declaró laurence d. fink, presidente y consejero delegado de blackrock

ECO:  
correcto:



ferrocarrils de la generalitat de catalunya (fgc) cerró 2021 con más de 61 millones de viajes, lo que supuso un aumento del 28% de la demanda respecto a 2020 y el 67% de la que hubo en 2019, según un comunicado de la empresa este viernes. la demanda fue en aumento durante el año y en noviembre y diciembre las líneas metropolitanas ya habían recuperado el 80% de los pasajeros de dos años atrás. la línea barcelona-valls alcanzó 44,6 millones de viajes, un 29,5% más que en 2020; mientras que la línea llobregat-anoia se situó en 16,3 millones, un 25% más. la línea lleida-la pobla de segor es la que mejor se ha recuperado, con un aumento interanual del 98% y el 82,5% de la demanda de 2019, con 207.377 pasajeros. por otro lado, fgc ha explicado que los usuarios han puntuado el servicio con 77 puntos sobre 100, el mejor dato histórico, con la puntualidad, la accesibilidad y la señalización como valores con mejor nota. el índice de control de calidad, que mide de forma objetiva el grado de cumplimiento del servicio real respecto al programado, fue del 98,2% en barcelona-valls, el 99% en llobregat-anoia y el 99,65% en lleida-la pobla de segor

incorrecto:



la ministra de defensa, margarita robles, ha pedido a todos los miembros del gobierno "hablar menos y trabajar más", preguntada por la reciente polémica protagonizada por el titular de consumo, alberto garzón, tras sus declaraciones sobre las macrogranjías a un medio de comunicación inglés. robles, que se ha expresado así este miércoles durante su visita a la brigada galicia vii brilat, en pontevedra, ha insistido en la necesidad de "seguir trabajando", dejando a un lado "opiniones personales". "a veces se habla mucho, hay que hablar menos y trabajar más", se ha mostrado además, en la línea de otros miembros del gobierno --incluido el propio presidente pedro sánchez--, partidaria de atajar la polémica cuanto antes, defendiendo el trabajo que hace el ejecutivo estatal. "el presidente, que era el único que se tenía que pronunciar, ya se ha pronunciado, así que lo que tenemos que hacer los demás, es trabajar", ha subrayado. "contradicciones en el pspe" por su parte, el ministro de consumo, alberto garzón, aseguraba este martes que ve "contradicciones" en el pspe sobre la cuestión de las macrogranjías, recordando que apoyó una moratoria del ejecutivo autonómico de castilla-la mancha en cuenca contra estos proyectos. el ministro reiteraba que su entrevista en el diario británico es "impecable" y se mostró convencido en "seguir esa línea de trabajo" respecto a este sector. garzón aclaró que la "ganadería extensiva, social y familiar" es sostenible y sirve para "proteger el territorio y arraigar población". a su juicio, este modelo "está amenazado por las macrogranjías", que "promueven el cambio climático", y afeó que el "lobby" cárnico "construyó un bulo en torno a sus palabras. además, aseguró que no se siente desautorizado por las el presidente del gobierno, que lamentó la polémica por sus declaraciones, en línea con lo declarado este miércoles por la titular de defensa

SAN:

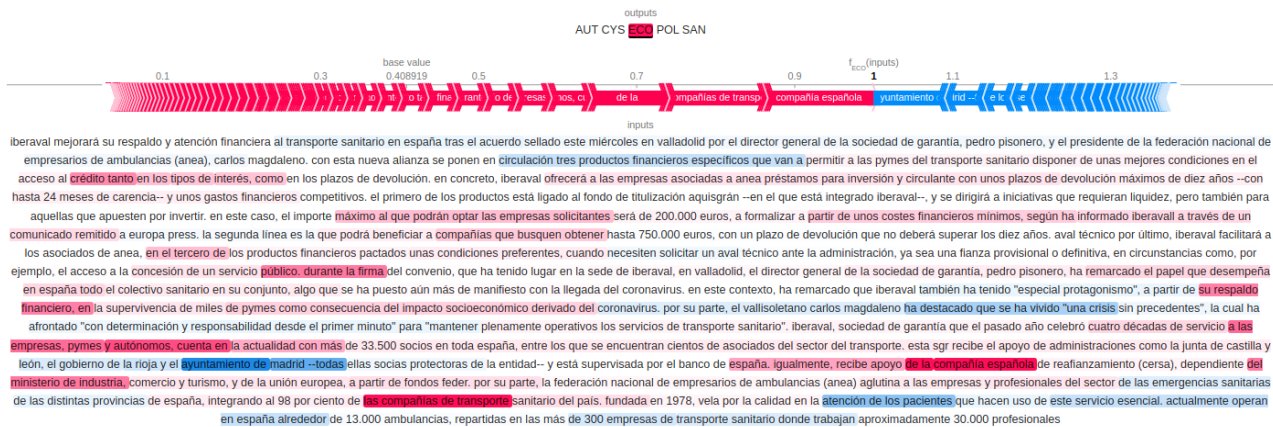
correcto:



el sindicato pide la destitución de la directora de enfermería por "saltarse a la torera" el orden de lista de profesionales ugt ha remitido un escrito a la gerencia del hospital sierrallana, tres mares en el que solicita la destitución de la directora de enfermería del centro hospitalario, rosa gema freire, por "incumplir y vulnerar" el protocolo establecido para el plan especial de vacunación, en el que los profesionales sanitarios administran vacunas fuera de su jornada laboral a cambio de una compensación económica. en nota de prensa, el sindicato recuerda que este plan, implantado por el gobierno de cantabria para aumentar la administración de dosis, implicaba que el personal de enfermería se inscribiera siempre de manera voluntaria en una lista para que luego se les fuera llamando por el orden establecido en un sorteo y de manera rotatoria, "lo que no se ha hecho en ningún momento", asegura. "a día de hoy hay profesionales a los que ya se les ha llamado dos veces y a otros ninguna e incluso se ha recurrido a trabajadores que ni siquiera estaban apuntados en la lista cuando a otros que no lo solicitaron en su momento no se les ha dejado incorporarse al plan especial de vacunación", según la sección sindical de ugt en sierrallana. el sindicato denuncia a la responsable de enfermería "por saltarse a la torera" el protocolo del plan en el hospital, a quien acusa de premiar especialmente "con jornadas de vacunación extraordinarias" a aquellos trabajadores que estén dispuestos a trabajar en sus días de descanso por necesidades del servicio. ugt critica que "no se puede pedir la colaboración voluntaria del personal sanitario para intensificar la campaña de vacunación para luego discriminar a unos y premiar a otros según los criterios o intereses personales de ciertas personas que nada tienen que ver con el protocolo establecido". el plan especial de vacunación del gobierno estipula una compensación económica de 300 euros para el personal facultativo y de 250 euros para el de enfermería por cinco horas de trabajo fuera de su jornada laboral

incorrecto:





## NER: cosas hechas/en proceso

Datasets para probar NER:

<https://metatext.io/datasets-list/ner-task>

- Ester: probar con spacy lo siguiente. Objetivo: ver cómo automatizar pruebas sobre NER
  - Sacar métricas (bondad en cuanto a la capacidad de detección, bondad en cuanto a la capacidad de clasificación de entidad) en dataset wikiner con modelo es\_core\_news\_lg (spacy, el modelo saca NER. Este NER de spacy no utiliza un transformer sino un tok2vec. Chequear si tok2vec trabaja a nivel de palabra)
  - Lo mismo pero con modelo xx\_ent\_wiki\_sm (spacy)
  - Buscar otros dataset
- Pedro: cogió noticias concretas en dataset 16k y chequeó entidades (modelo es\_core\_news\_lg [spacy]). Conclusiones: a ojo, spacy saca las entidades bien. También hizo tf-idf enfatizando las entidades extraídas por NER y luego sacaba LDA. Conclusiones: saca prácticamente lo mismo que con el LDA sin NER.
- Pedro: cogió dataset wikiner y chequeó entidades (modelo es\_core\_news\_lg [spacy] y modelo 'es-ner-large' [flair]). Para evaluarlo se ha estado estudiando cómo medir el desempeño de los modelos NER, tanto qué métricas usar como el cómo comparar dos entidades (si a nivel de token, a nivel de entidad completa, teniendo en cuenta el overlap entre la entidad real y la predicha...).

Las conclusiones que se han sacado han sido que las métricas más usadas son:

1. Recall
2. Precision
3. F1-Score

Además, se han considerado otras métricas como 'accuracy' y 'jaccard'.

- El 'accuracy' cuenta los Verdaderos Negativos (TN) que no son especialmente interesantes para el problema, ya que, en el caso de considerar las métricas token a token, muchas 'stopwords' y palabras

comunes serán tokens que no serán entidades y se considerarán TN, ocultando el desempeño sobre las palabras realmente importantes.

- El 'jaccard' propone un 'Intersection over Union'. Es una métrica que tiene en cuenta lo mismo que el f1-score, siendo más exigente que ésta. Por lo tanto dará una información similar a la del f1-score.

Para la comparación entre entidades se ha encontrado esta fuente:

[https://www.davidsbatista.net/blog/2018/05/09/Named\\_Entity\\_Evaluation/](https://www.davidsbatista.net/blog/2018/05/09/Named_Entity_Evaluation/)

dentro de este enlace hay referencias a distintos githubs y papers

papers:

1. <https://www.semanticscholar.org/paper/The-Automatic-Content-Extraction-%28ACE%29-Program-and-Doddington-Mitchell/0617dd6924df7a3491c299772b70e90507b195dc?p2df>
2. <https://aclanthology.org/M93-1007/>
  - a. github: <https://github.com/jantrienes/nereval>

github general: <https://github.com/davidsbatista/NER-Evaluation>

Describe varias alternativas. La que se ha usado hasta ahora en WikiNER ha sido una variante del tipo **Partial** (partial boundary match over the surface string, regardless of the type) [International Workshop on Semantic Evaluation (SemEval)]. En el tipo **Partial** considera que, todas aquellas entidades que se han detectado y que se superponen (en cuanto a texto) con las reales, se tienen que evaluar como si fuera media entidad correcta (es decir, se cuenta como correcta y se multiplica por 0.5)

Así, se ha medido *a nivel de token*: dividiendo cada entidad por token y asignándole a ese token el tipo de entidad de la que viene. Se evaluará tanto la detección como la clasificación. La evaluación de la clasificación se hará sobre los tokens de las entidades bien detectadas, es decir, sobre los TP de la detección. Resultados :

1. Spacy: (Notebook en máquina virtual mv1-europapress: /pedro/NER/NER\_test.ipynb [test\_v0] VERSIÓN 0 sobre wikiner):

En esta versión se procesa el texto a la vez que se va testeando. Para cada texto se sacan sus entidades. Por cada entidad predicha se va avanzando en los tokens del texto hasta llegar a un token perteneciente a esta entidad predicha. Este método permite, por lo tanto, tener en cuenta los **True Negative**, lo que implica poder medir el *accuracy*.

Los resultados de esta función (Spacy: (Notebook en máquina virtual mv1-europapress: /pedro/NER/NER\_test.ipynb [test\_v0] VERSIÓN 0 sobre wikiner-wp3)) son:



WIKINER - SPACY - test_v0					
	DETECCIÓN	CLASIFICACIÓN			
		LOC	MISC	ORG	PER
ACCURACY	0.9439	0.9898	0.985	0.9808	0.9815
RECALL	0.9633	0.9788	0.922	0.9731	0.9018
PRECISION	0.979	0.9791	0.9448	0.9448	0.9004
F1-SCORE	0.9712	0.978	0.9327	0.9333	0.9011
JACCARD	0.9439	0.957	0.8749	0.9406	0.82

Sin embargo, tiene problemas, ya que podría darse el caso de que un token del texto coincida con el de la entidad predicha, pero no sea justamente el token de la entidad, es decir, es una palabra repetida en el texto y por lo tanto puede causar problemas. Ejemplo:

“Las maderas de Maderas SL son muy buenas”

Entidad real: “Maderas SL” (ORG)

Entidad detectada: “Maderas SL” (ORG)

Leemos por cada token:

“las” in “maderas sl”  $\Rightarrow$  FALSE

“maderas” in “maderas sl”  $\Rightarrow$  TRUE

“maderas” (token nº2) no pertenece a una entidad real por lo tanto sería un error de detección. Se pasaría a la siguiente entidad detectada.

¿Cómo estaría bien? Teniendo en cuenta la posición de la entidad y el token  
Nota: usaremos el operador < para indicar que la cadena de la izquierda del operador va delante de la cadena de la derecha. El operador > al revés, y se usará el operador  $\wedge$  para indicar que una está contenida en otra (hay intersección real).

“las”  $\wedge$  “maderas sl”  $\Rightarrow$  FALSE (porque “las” < “maderas sl”)

“maderas”  $\wedge$  “maderas sl”  $\Rightarrow$  FALSE (porque “maderas” < “maderas sl”)

“de”  $\wedge$  “maderas sl”  $\Rightarrow$  FALSE (porque “de” < “maderas sl”)

“maderas”  $\wedge$  “maderas sl”  $\Rightarrow$  TRUE

“maderas” (token nº4) es una entidad real y por lo tanto está bien detectada.

“sl”  $\wedge$  “maderas sl”  $\Rightarrow$  TRUE

“sl” (token nº5) es una entidad real y por lo tanto está bien detectada.

Además, hay que calcular las entidades en cada ejecución lo que provoca que la ejecución sea lenta.

**Ventajas:**

- Detección de True Negatives y medición de accuracy.

**Problemas:**

- Palabras comunes repetidas en el texto pueden dar problemas.
- Mucho tiempo para testear nuevas métricas.

**Solución:**

- Utilizar offset dentro del texto para localizar las entidades
- Guardar en disco las entidades detectadas y las reales.

1. (Notebook en máquina virtual mv1-europapress:  
/pedro/NER/NER\_test.ipynb [test\_v1] VERSIÓN 1):

Las mejoras propuestas en la versión anterior han sido utilizadas en esta nueva versión. Además no se tienen en cuenta las stopwords dadas por NLTK.

**Ventajas:**

- Velocidad: entidades detectadas y reales precalculadas.

**Problemas:**

- No se pueden tener en cuenta los True Negative en la detección.

Para localizar las entidades a través del offset se ha utilizado como primera aproximación la posición del token dentro del texto. Tanto *Spacy* como *Flair* utilizan su propio tokenizador para procesar el texto. Los resultados obtenidos en *Flair* y *Spacy* son:

CoNLL - SPACY - test_v1 (token offset)					
	DETECCIÓN	CLASIFICACIÓN			
		LOC	MISC	ORG	PER
RECALL	0.9298	0.8971	0.5309	0.6403	0.9045
PRECISION	0.8189	0.5672	0.5193	0.8725	0.9184

F1-SCORE	0.8709	0.6951	0.5251	0.7385	0.9114
JACCARD	0.7713	0.5326	0.356	0.5855	0.8372

	WIKINER - SPACY - test_v1 (token offset)				
	DETECCIÓN	CLASIFICACIÓN			
		LOC	MISC	ORG	PER
RECALL	0.9797	0.9744	0.9228	0.907	0.9802
PRECISION	0.9872	0.9673	0.9452	0.91	0.9775
F1-SCORE	0.9834	0.9708	0.9339	0.9085	0.9789
JACCARD	0.9674	0.9433	0.876	0.8324	0.9586

2. Flair: saca resultados algo peores:

	CoNLL - FLAIR - test_v1 (token offset)				
	DETECCIÓN	CLASIFICACIÓN			
		LOC	MISC	ORG	PER
RECALL	0.9729	0.927	0.9559	0.9707	0.9984
PRECISION	0.9916	0.9558	0.9615	0.9601	0.9923
F1-SCORE	0.9821	0.9412	0.9587	0.9654	0.9954
JACCARD	0.9649	0.8889	0.9207	0.9330	0.9907

	WIKINER - FLAIR - test_v1 (token offset)				
	DETECCIÓN	CLASIFICACIÓN			
		LOC	MISC	ORG	PER
RECALL	0.963	0.8108	0.8627	0.8513	0.9521

PRECISION	0.9726	0.9605	0.7154	0.6096	0.9671
F1-SCORE	0.9678	0.8793	0.7821	0.7104	0.9596
JACCARD	0.9376	0.7846	0.6422	0.5509	0.9223

Tomar el offset del token puede dar problemas al no usar un tokenizador común. Además, al incluir en el estudio el NER de Google Cloud es necesario utilizar un offset a nivel de carácter. Es por ello que se reformuló la solución utilizando el offset a nivel de char tanto en Spacy, Flair y Google Cloud.

1. Google Cloud: este NER tiene en cuenta muchas más entidades, concretamente tiene en cuenta entidades del tipo:

UNKNOWN	Unknown
PERSON	Person
LOCATION	Location
ORGANIZATION	Organization
EVENT	Event
WORK_OF_ART	Artwork
CONSUMER_GOOD	Consumer product
OTHER	Other types of entities

PHONE_NUMBER	<p>Phone number</p> <p>The metadata lists the phone number, formatted according to local convention, plus whichever additional elements appear in the text:</p> <ul style="list-style-type: none"> <li>• number - the actual number, broken down into sections as per local convention</li> <li>• national_prefix - country code, if detected</li> <li>• area_code - region or area code, if detected</li> <li>• extension - phone extension (to be dialed after connection), if detected</li> </ul>
ADDRESS	<p>Address</p> <p>The metadata identifies the street number and locality plus whichever additional elements appear in the text:</p> <ul style="list-style-type: none"> <li>• street_number - street number</li> <li>• locality - city or town</li> <li>• street_name - street/route name, if detected</li> <li>• postal_code - postal code, if detected</li> <li>• country - country, if detected</li> <li>• broad_region - administrative area, such as the state, if detected</li> <li>• narrow_region - smaller administrative area, such as county, if detected</li> <li>• sublocality - used in Asian addresses to demark a district within a city, if detected</li> </ul>
DATE	<p>Date</p> <p>The metadata identifies the components of the date:</p> <ul style="list-style-type: none"> <li>• year - four digit year, if detected</li> <li>• month - two digit month number, if detected</li> <li>• day - two digit day number, if detected</li> </ul>

NUMBER	Number  The metadata is the number itself.
PRICE	Price  The metadata identifies the value and currency.

Así, una primera estrategia es encasillar todos los tipos distintos de UNKNOWN, PERSON, ORGANIZATION y LOCATION como MISCELÁNEA.

Así, los resultados han sido:

CoNLL - Google Cloud - test_v1 (char offset)					
	DETECCIÓN	CLASIFICACIÓN			
		LOC	MISC	ORG	PER
RECALL	0.9487	0.8678	0.5762	0.7754	0.951
PRECISION	0.332	0.6985	0.6373	0.8838	0.904
F1-SCORE	0.4919	0.774	0.6052	0.8261	0.9269
JACCARD	0.3262	0.6313	0.4339	0.7037	0.8638

WIKINER - Google Cloud - test_v1 (char offset)					
	DETECCIÓN	CLASIFICACIÓN			
		LOC	MISC	ORG	PER
RECALL	0.9903	0.8726	0.7125	0.7908	0.9062
PRECISION	0.3257	0.9269	0.7555	0.5868	0.9052
F1-SCORE	0.4902	0.899	0.7333	0.6737	0.9057
JACCARD	0.3247	0.8165	0.579	0.508	0.8277



2. Spacy:

CoNLL - SPACY - test_v1 (char offset)					
	DETECCIÓN	CLASIFICACIÓN			
		LOC	MISC	ORG	PER
RECALL	0.9315	0.8971	0.531	0.64	0.9045
PRECISION	0.8202	0.5673	0.5193	0.8724	0.9184
F1-SCORE	0.8723	0.6951	0.5251	0.7384	0.9114
JACCARD	0.7736	0.5326	0.356	0.5853	0.8372

WIKINER - SPACY - test_v1 (char offset)					
	DETECCIÓN	CLASIFICACIÓN			
		LOC	MISC	ORG	PER
RECALL	0.9822	0.9744	0.9228	0.907	0.9802
PRECISION	0.9894	0.9673	0.9452	0.91	0.9775
F1-SCORE	0.9858	0.9708	0.9339	0.9085	0.9789
JACCARD	0.9719	0.9433	0.876	0.8324	0.9586

3. Flair:

CoNLL - FLAIR - test_v1 (char offset)					
---------------------------------------	--	--	--	--	--

	DETECCIÓN	CLASIFICACIÓN			
		LOC	MISC	ORG	PER
RECALL	0.9747	0.927	0.9559	0.9707	0.9984
PRECISION	0.994	0.9558	0.9615	0.9601	0.9923
F1-SCORE	0.9842	0.9412	0.9412	0.9654	0.9954
JACCARD	0.9689	0.8889	0.8889	0.933	0.9907

WIKINER - FLAIR - test_v1 (char offset)					
	DETECCIÓN	CLASIFICACIÓN			
		LOC	MISC	ORG	PER
RECALL	0.9652	0.8108	0.8627	0.8513	0.9521
PRECISION	0.9752	0.9605	0.7154	0.6096	0.9671
F1-SCORE	0.9702	0.8793	0.7822	0.7104	0.9596
JACCARD	0.9421	0.7846	0.6423	0.5509	0.9223

Precisamente la clase MISCELÁNEA es problemática (principalmente en Google Cloud) así, se ha decidido testear los datasets eliminando las entidades de esta clase. Rehacemos las tablas sin tener en cuenta este tipo:

#### 1. Google Cloud:

CoNLL - Google Cloud - test_v1 (char offset) [MISC]				
	DETECCIÓN	CLASIFICACIÓN		
		LOC	ORG	PER
RECALL	0.9319	0.8989	0.8391	0.9714

PRECISION	0.6248	0.7525	0.9545	0.9424
F1-SCORE	0.7479	0.8192	0.8931	0.9567
JACCARD	0.5973	0.6938	0.8068	0.917

WIKINER - Google Cloud - test_v1 (char offset) [MISC]				
	DETECCIÓN	CLASIFICACIÓN		
		LOC	ORG	PER
RECALL	0.9694	0.9052	0.868	0.9457
PRECISION	0.6014	0.9546	0.707	0.9418
F1-SCORE	0.7223	0.9293	0.7793	0.9438
JACCARD	0.8679	0.8679	0.6383	0.8935

## 2. Spacy

CoNLL - SPACY - test_v1 (char offset) [MISC]				
	DETECCIÓN	CLASIFICACIÓN		
		LOC	ORG	PER
RECALL	0.9061	0.9272	0.7136	0.9354
PRECISION	0.9351	0.6114	0.9536	0.945
F1-SCORE	0.9204	0.7369	0.8163	0.9402
JACCARD	0.8525	0.5834	0.6897	0.6897

WIKINER - SPACY - test_v1 (char offset) [MISC]				
		CLASIFICACIÓN		

	DETECCIÓN	LOC	ORG	PER
RECALL	0.9825	0.9805	0.9434	0.9863
PRECISION	0.9882	0.9805	0.9415	0.987
F1-SCORE	0.9854	0.9805	0.9425	0.9866
JACCARD	0.9711	0.9618	0.8912	0.9736

### 3. Flair

CoNLL - FLAIR - test_v1 (char offset) [MISC]				
	DETECCIÓN	CLASIFICACIÓN		
		LOC	ORG	PER
RECALL	0.9785	0.931	0.9826	0.9984
PRECISION	0.9898	0.9622	0.9697	0.9966
F1-SCORE	0.9841	0.9463	0.9761	0.9975
JACCARD	0.9688	0.8981	0.9534	0.995

WIKINER - FLAIR - test_v1 (char offset) [MISC]				
	DETECCIÓN	CLASIFICACIÓN		
		LOC	ORG	PER
RECALL	0.9528	0.8831	0.9507	0.9785
PRECISION	0.9865	0.9798	0.6553	0.9823
F1-SCORE	0.9694	0.929	0.7758	0.9804
JACCARD	0.9406	0.8673	0.6337	0.9616

---

## TABLAS COMPARATIVAS.

---

### DETECCIÓN

	CoNLL - DETECCIÓN - test_v1 (char offset) [MISC]		
	Google Cloud	SPACY	FLAIR
RECALL	0.9312	0.9061	0.9785
PRECISION	0.6249	0.9351	0.9898
F1-SCORE	0.7479	0.9204	0.9841
JACCARD	0.5973	0.8525	0.9688

	WIKINER - DETECCIÓN - test_v1 (char offset) [MISC]		
	Google Cloud	SPACY	FLAIR
RECALL	0.9694	0.9825	0.9528
PRECISION	0.6014	0.9882	0.9865
F1-SCORE	0.7423	0.9854	0.9694
JACCARD	0.5901	0.9711	0.9406

---

## CLASIFICACIÓN.

---

### LOCATION

	CoNLL - CLASIFICACIÓN LOC - test_v1 (char offset) [MISC]		
	Google Cloud	SPACY	FLAIR
RECALL	0.8989	0.9273	0.9785
PRECISION	0.7525	0.6114	0.9898
F1-SCORE	0.8192	0.7369	0.9463
JACCARD	0.6938	0.5835	0.8981

WIKINER - CLASIFICACIÓN LOC - test_v1 (char offset) [MISC]			
	Google Cloud	SPACY	FLAIR
RECALL	0.9052	0.9805	0.8831
PRECISION	0.9546	0.9805	0.9798
F1-SCORE	0.9293	0.9805	0.929
JACCARD	0.8679	0.9618	0.8673

---

## ORGANIZATION

CoNLL - CLASIFICACIÓN ORG - test_v1 (char offset) [MISC]			
	Google Cloud	SPACY	FLAIR
RECALL	0.8391	0.7136	0.9826
PRECISION	0.9545	0.9536	0.9697
F1-SCORE	0.8931	0.8163	0.9761
JACCARD	0.8068	0.6897	0.9534

WIKINER - CLASIFICACIÓN ORG - test_v1 (char offset)[MISC]			
	Google Cloud	SPACY	FLAIR
RECALL	0.8680	0.9434	0.9507
PRECISION	0.7069	0.9415	0.6553
F1-SCORE	0.7793	0.9425	0.7758
JACCARD	0.6384	0.8912	0.6337



---

## PERSON

CoNLL - CLASIFICACIÓN PER - test_v1 (char offset) [MISC]			
	Google Cloud	SPACY	FLAIR
RECALL	0.9714	0.9354	0.9984
PRECISION	0.9424	0.945	0.9966
F1-SCORE	0.9567	0.9402	0.9975
JACCARD	0.917	0.8871	0.9950

WIKINER - CLASIFICACIÓN PER - test_v1 (char offset) [MISC]			
	Google Cloud	SPACY	FLAIR
RECALL	0.9457	0.9863	0.9785
PRECISION	0.9418	0.9869	0.9823
F1-SCORE	0.9438	0.9866	0.9804
JACCARD	0.8935	0.9736	0.9616

En cuanto a tiempos, la detección de entidades que menos tarda es la de SPACY. Poniendo este modelo como base, Google Cloud tarda sobre 5 veces más que SPACY y FLAIR tarda sobre 20 veces más que SPACY.

## AUTOENCODERS PARA REDUCIR DIMENSIONALIDAD

Se va a estudiar la capacidad de los autoencoders para reducir la dimensionalidad desde las 512 dimensiones que tenemos originalmente. Los resultados dados por los autoencoders se contrastarán con los correspondientes modelos de PCA y U-MAP. Para ello, se tiene que elegir una dimensión K1 a la que reducir el espacio. Para generalizar se van a estudiar los valores

K1 in {16,32,64,128}

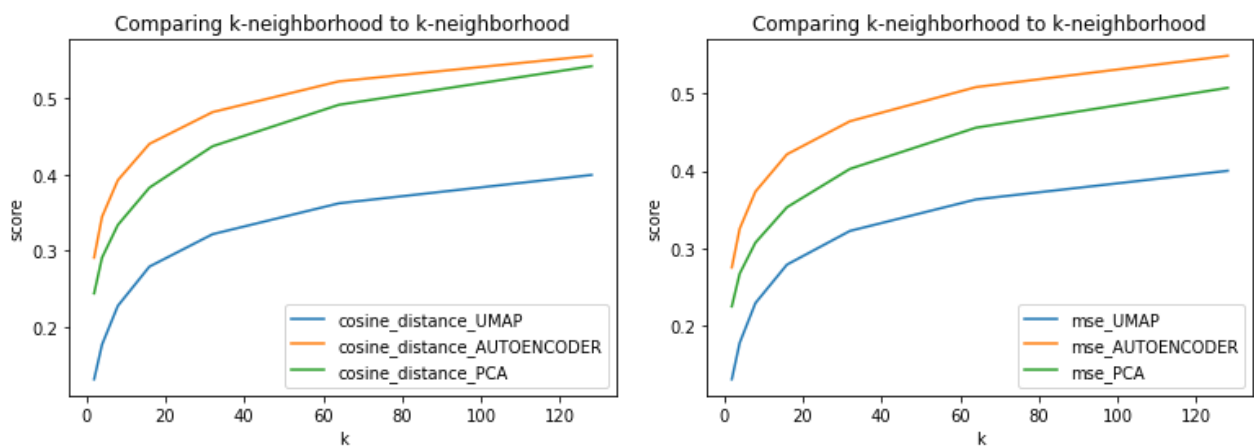
Como requisito para un buen desempeño de los modelos al reducir dimensionalidad se necesita que cada instancia conserve su vecindario. El vecindario estará formado por los K2 vecinos más cercanos. Para generalizar también vamos a mover el K2 sobre los valores {2,4,8,16,32,64,128}. Además, una buena reducción hará que un clasificador conserve su clasificación del espacio original en el reducido. Así, haremos varios estudios:

1. Qué porcentaje del K2-vecindario de dimensión original se conserva en el K2-vecindario del espacio reducido para cada instancia.
2. Dado un porcentaje X, ¿qué valor de K2' se necesita para que el K2'-vecindario del espacio reducido mantenga un X% del K2-vecindario original?
3. ¿Se conserva la clasificación? Clasificaremos las instancias tanto en el espacio original como en el espacio reducido. Compararemos las dos clasificaciones para ver cuánto se han conservado.

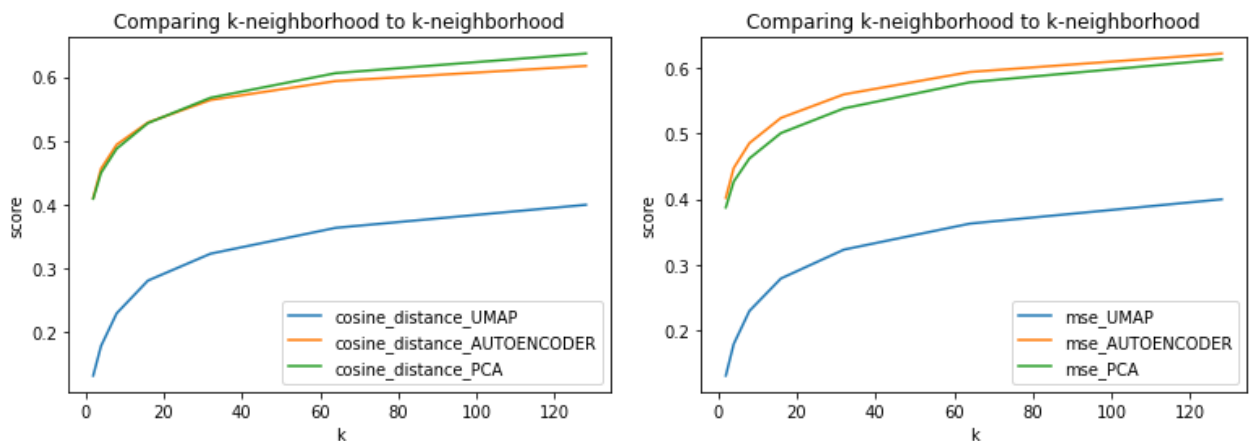
# 1. Qué porcentaje del K2-vecindario de dimensión original se conserva en el K2-vecindario del espacio reducido para cada instancia.

Vemos los resultados para cada K1-espacio.

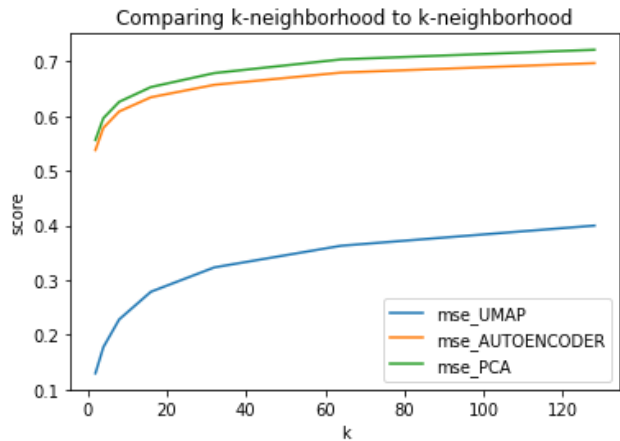
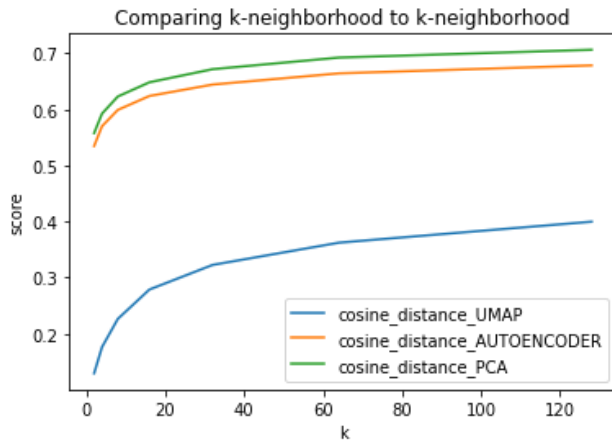
## • K1=16



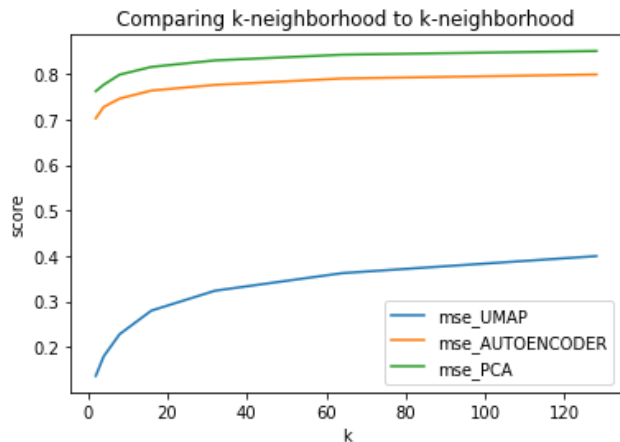
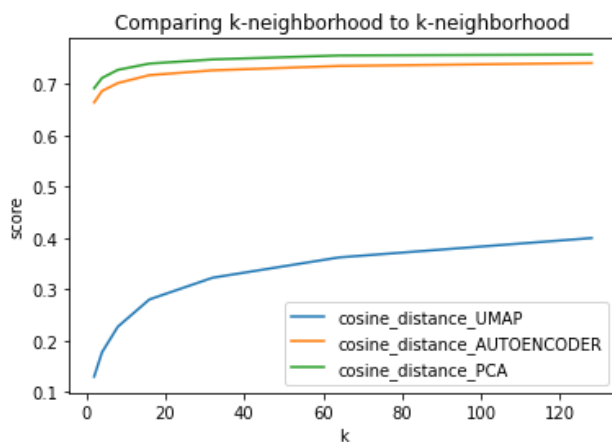
## • K1=32



- K1=64



- K1=128



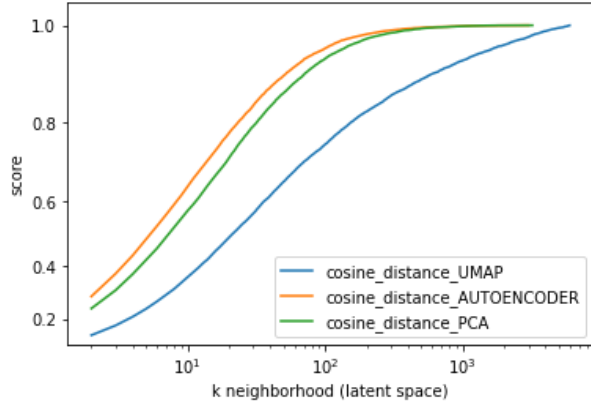
En este primer estudio resalta el desempeño de U-MAP, que se queda lejos de conseguir los resultados de sus competidores. Por otro lado, entre el AutoEncoder y PCA tenemos resultados similares. Para K2 más pequeños el autoencoder mejora a PCA. Sin embargo, para un K2 mayor (por ejemplo de 64 y 128) es PCA quien consigue mejores resultados. Esto va a ayudar a concluir que cuando se requiere una reducción de dimensionalidad a un número muy pequeño de características es mejor opción el autoencoder.

## 2. Dado un porcentaje X, ¿qué valor de K2' se necesita para que el K2'-vecindario del espacio reducido mantenga un X% del K2-vecindario original?

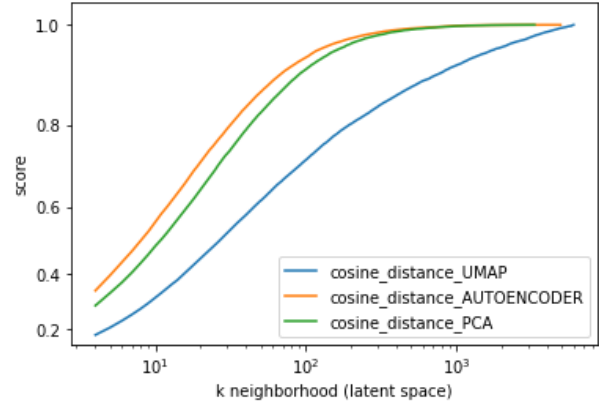
Para este apartado se va a fijar el valor de K2 (número de vecinos en el espacio original) y nos vamos a mover con K2' (número de vecinos en el espacio reducido). En este caso se van a mostrar los valores para los K1-espacio (espacio reducido de dimensión K1) con K1=16 y K1=128.

- K1=16 y distancia coseno

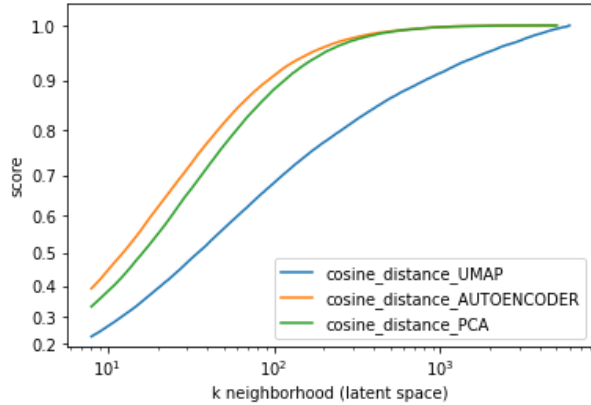
Comparing 2(fijo)-neighborhood to (2==>inf)-neighborhood



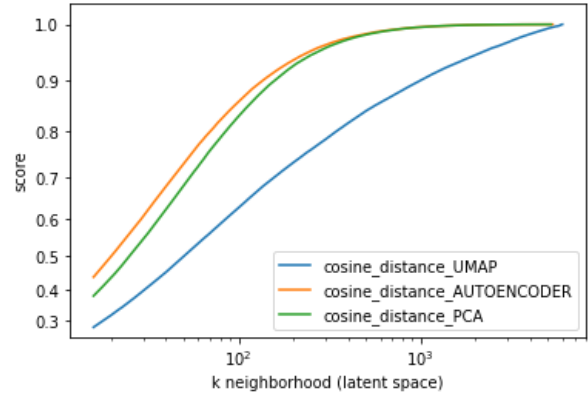
Comparing 4(fijo)-neighborhood to (4==>inf)-neighborhood



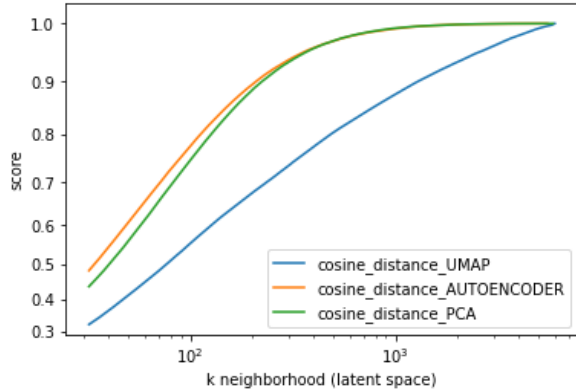
Comparing 8(fijo)-neighborhood to (8==>inf)-neighborhood



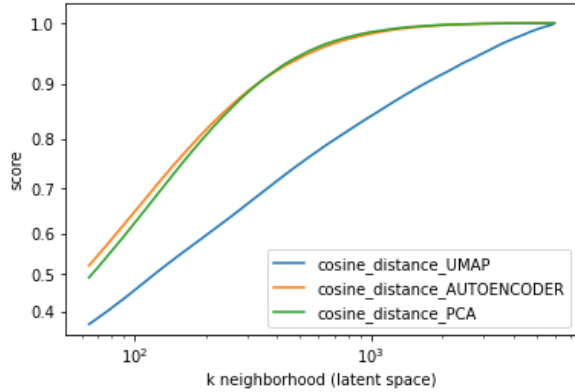
Comparing 16(fijo)-neighborhood to (16==>inf)-neighborhood



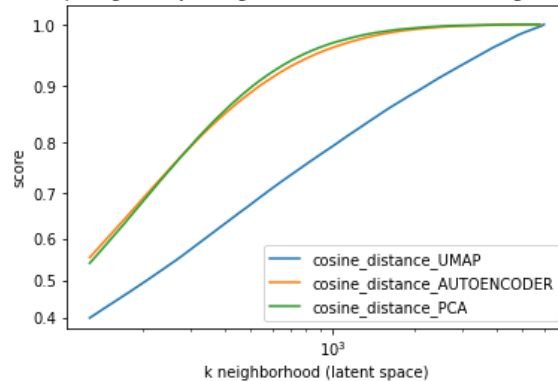
Comparing 32(fijo)-neighborhood to (32==>inf)-neighborhood



Comparing 64(fijo)-neighborhood to (64==>inf)-neighborhood

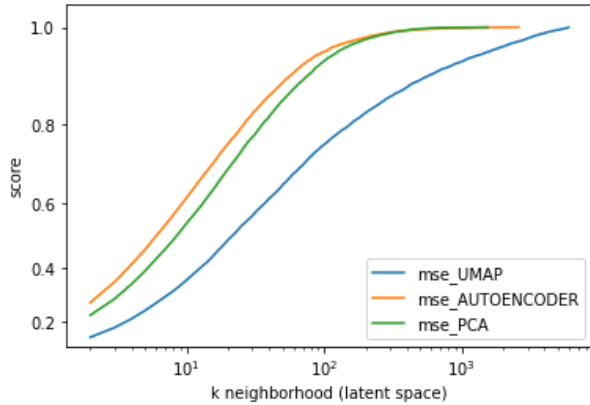


Comparing 128(fijo)-neighborhood to (128==>inf)-neighborhood

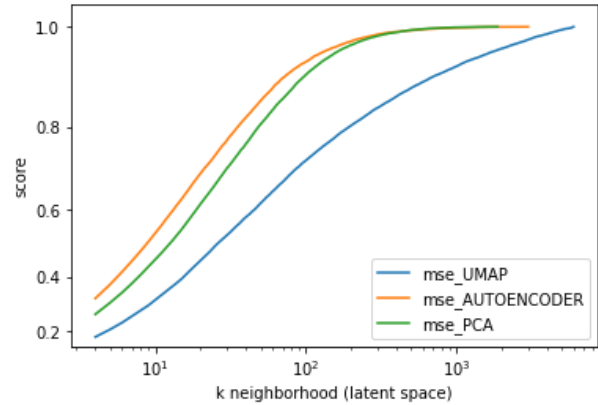


- K1=16 y MSE

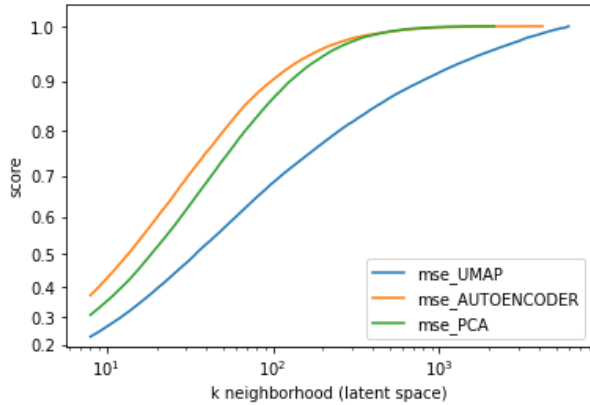
Comparing 2(fijo)-neighborhood to (2==>inf)-neighborhood



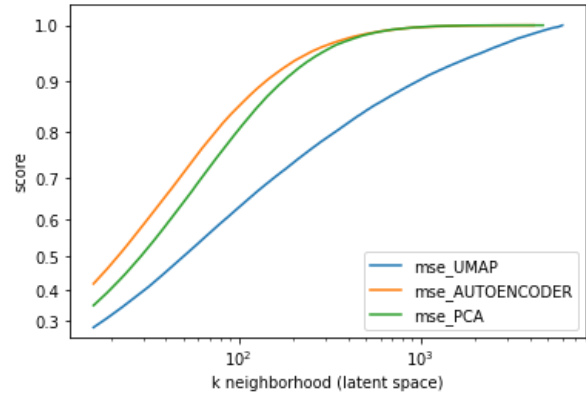
Comparing 4(fijo)-neighborhood to (4==>inf)-neighborhood



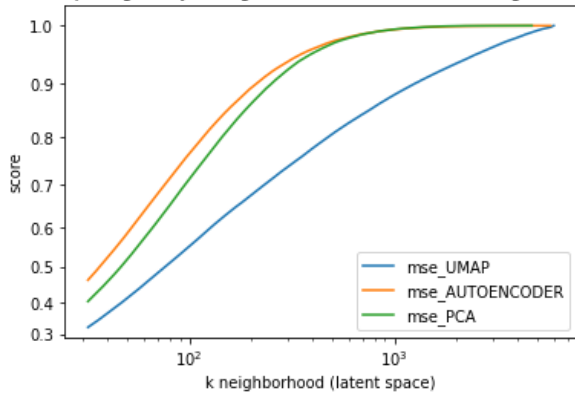
Comparing 8(fijo)-neighborhood to (8==>inf)-neighborhood



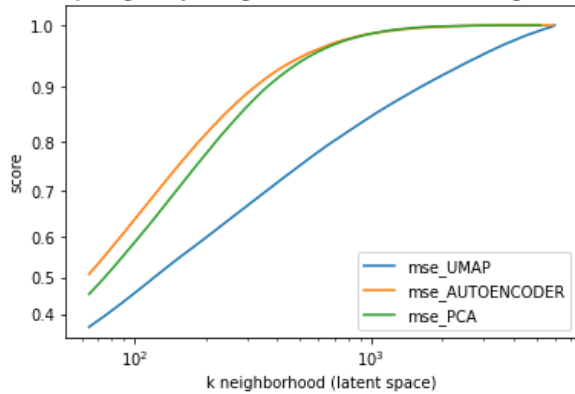
Comparing 16(fijo)-neighborhood to (16==>inf)-neighborhood



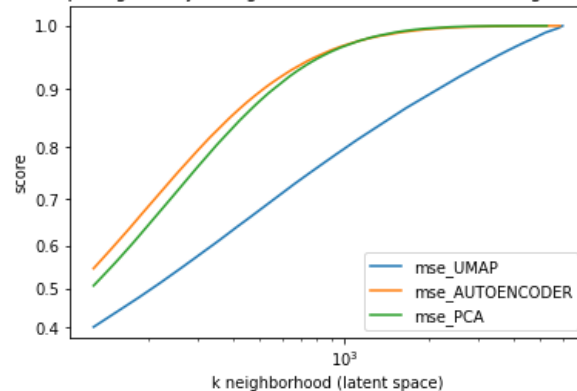
Comparing 32(fijo)-neighborhood to (32==>inf)-neighborhood



Comparing 64(fijo)-neighborhood to (64==>inf)-neighborhood

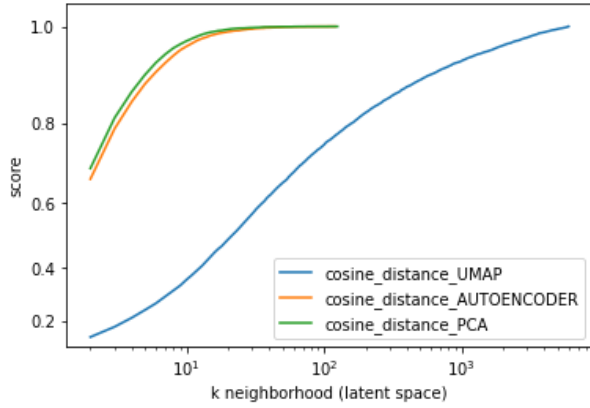


Comparing 128(fijo)-neighborhood to (128==>inf)-neighborhood

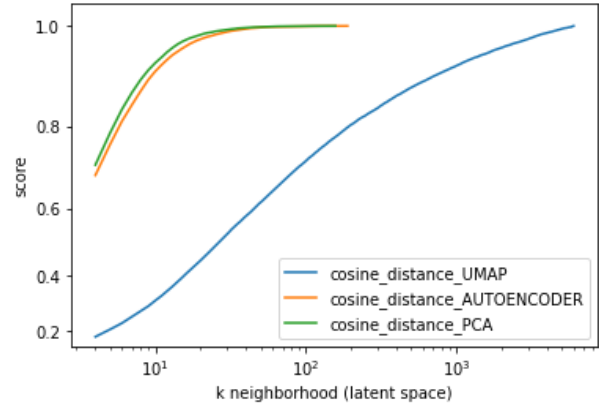


- K1=128 y distancia coseno

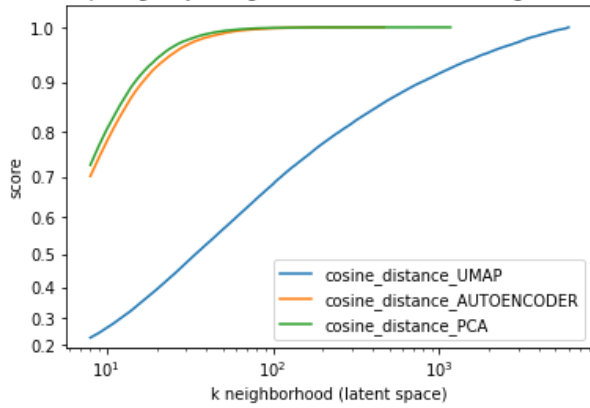
Comparing 2(fijo)-neighborhood to (2==>inf)-neighborhood



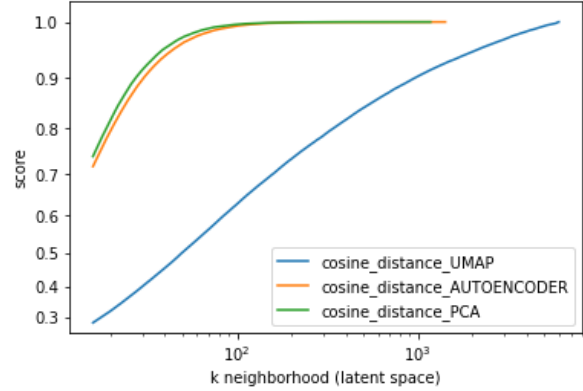
Comparing 4(fijo)-neighborhood to (4==>inf)-neighborhood



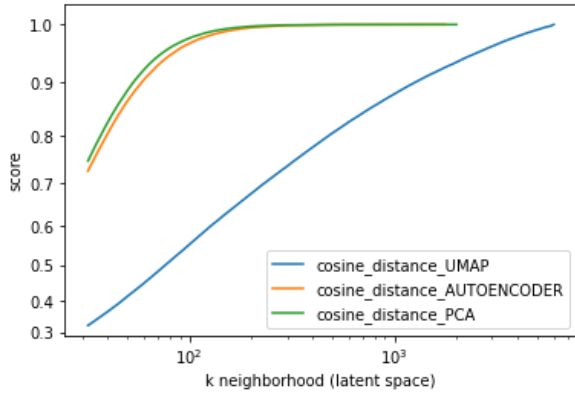
Comparing 8(fijo)-neighborhood to (8==>inf)-neighborhood



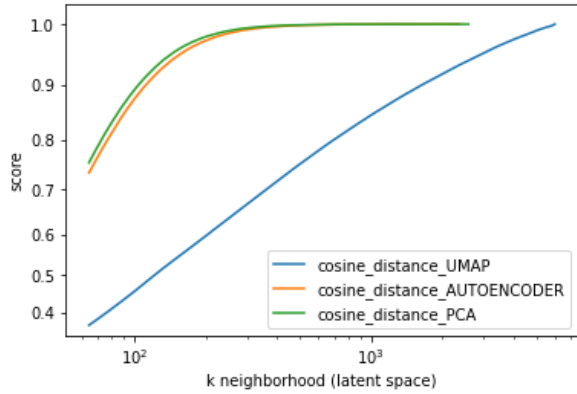
Comparing 16(fijo)-neighborhood to (16==>inf)-neighborhood



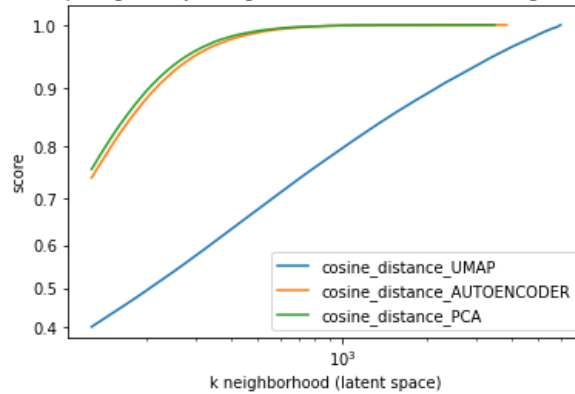
Comparing 32(fijo)-neighborhood to (32==>inf)-neighborhood



Comparing 64(fijo)-neighborhood to (64==>inf)-neighborhood



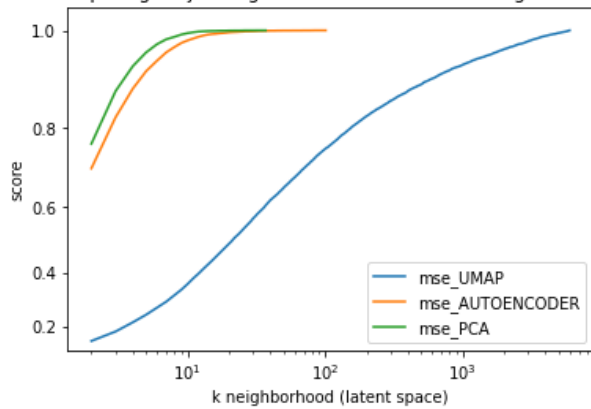
Comparing 128(fijo)-neighborhood to (128==>inf)-neighborhood



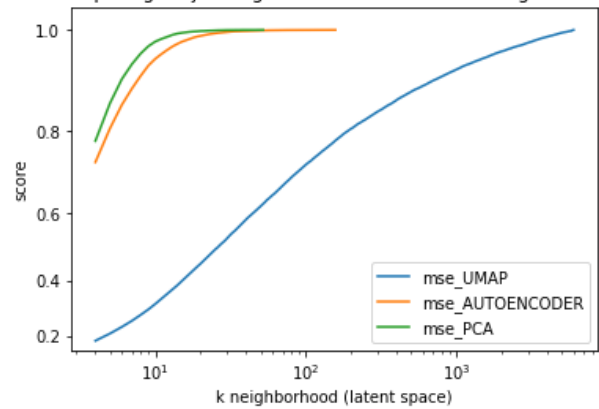


- K1=128 y MSE

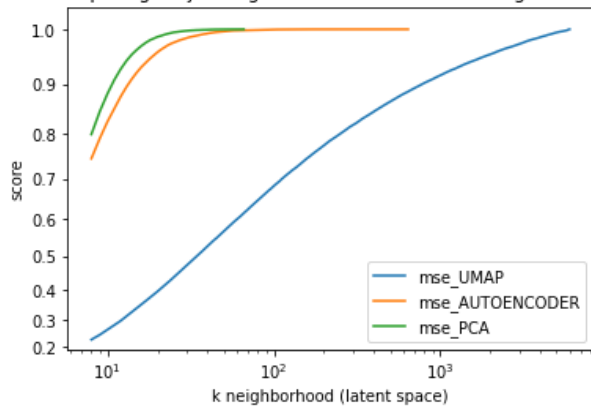
Comparing 2(fijo)-neighborhood to (2==>inf)-neighborhood



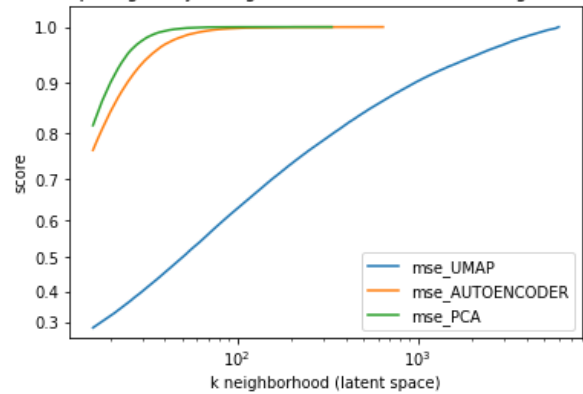
Comparing 4(fijo)-neighborhood to (4==>inf)-neighborhood



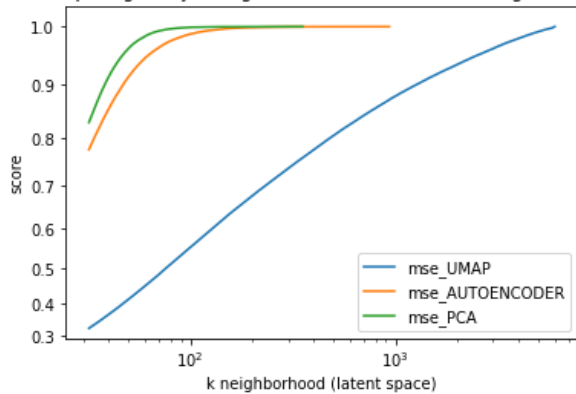
Comparing 8(fijo)-neighborhood to (8==>inf)-neighborhood



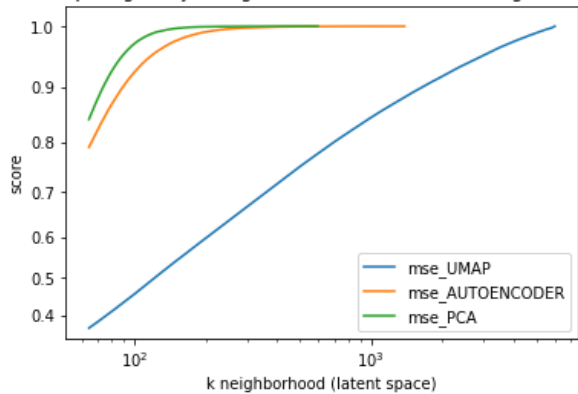
Comparing 16(fijo)-neighborhood to (16==>inf)-neighborhood



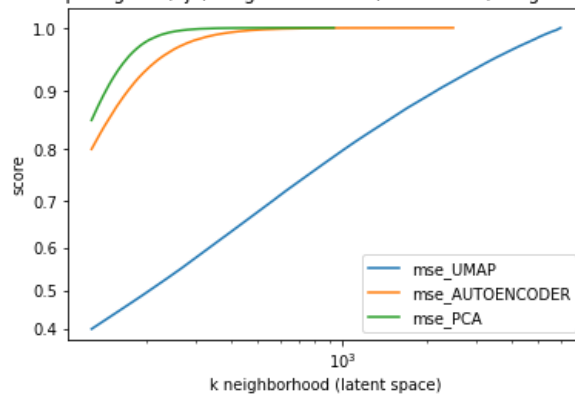
Comparing 32(fijo)-neighborhood to (32==>inf)-neighborhood



Comparing 64(fijo)-neighborhood to (64==>inf)-neighborhood



Comparing 128(fijo)-neighborhood to (128==>inf)-neighborhood



Para un K1=16 los modelos de autoencoder mejoran a PCA, sin embargo para un K1=128 pasa al contrario. Volviendo a tener una conclusión parecida al apartado anterior. Además, se puede observar como la distancia coseno beneficia al autoencoder en el 128-espacio.

3. ¿Se conserva la clasificación? Clasificaremos las instancias tanto en el espacio original como en el espacio reducido. Compararemos las dos clasificaciones para ver cuánto se han conservado.

Para este apartado se han utilizado dos modelos:

- Random Forest
- Regresión Logística

Se han escogido las 6 clases más representativas de los 20000 documentos escogidos para esta prueba:

CYS 5670  
POL 3744  
AUT 3549  
ECO 2894  
SAN 1777  
CUL 1005  
TRI 530  
DEP 517  
MOT 144  
INV 104  
EDU 57  
OCI 6  
PRT 3

Estas clases son ['CYS', 'POL', 'AUT', 'ECO', 'SAN', 'CUL']. Sobre estas clases se ha entrenado un modelo de Random Forest en distintos espacios de dimensionalidad. Esta tabla representa los resultados del accuracy en test para cada dimensión:

RANDOM FOREST					
N-SPACE (ORIGINAL = 0.6815)					
	2	4	8	16	32
PCA	0.3899	0.5795	0.6196	0.6702	0.6794
AUTOENCODER	0.5354	0.6219	0.6655	0.6738	0.6758

---

REGRESIÓN LOGÍSTICA

	N-SPACE (ORIGINAL = 0.6961)				
	2	4	8	16	32
PCA	0.4303	0.575	0.5892	0.6461	0.6686
AUTOENCODER	0.4339	0.5835	0.6196	0.6528	0.6688

La conclusión en este apartado está en la línea de lo que estábamos viendo en los anteriores. Cuando la reducción de dimensionalidad es muy grande (reducimos a una dimensión muy pequeña) el autoencoder sigue comportándose mejor, conservando gran parte de su clasificación original.

Juntando los resultados podemos concluir que PCA tiene un buen desempeño a la hora de reducir dimensionalidad en este problema hasta cierta dimensión, donde empieza a verse superado por el Autoencoder. Esto implica que nuestras características tienen cierta componente lineal y muchas de estas características tienen una alta correlación. Así, el AutoEncoder es un modelo capaz de detectar tanto estas dependencias lineales como otras más complejas, permitiendo reducir la dimensión hasta valores muy pequeños conservando gran parte de la información que PCA no es capaz de conservar.

**1. Cómo se haría el entrenamiento del NER y argumentar por qué no se haría (computacionalmente muy costosos, los datasets que tenemos quizás son muy pequeños para entrenar) (referencias y links)** Para entrenar un NER hace falta dataset etiquetado de gran volumen

Lo que hemos encontrado disponible en internet no es suficiente

Etiquetar nuestro dataset es costosísimo a nivel de tiempo y recursos

Como podemos adaptar modelos NER ya preentrenados, a nuestro problema, tiramos por aquí.

- Fran: pruebas de NER no supervisados sobre wikiner (Jacard score, “intersection over union” IoU, 42%).

entidad real: “Torrejón de Ardoz”. Detectado: “en Torrejón de Ardoz”. IoU: 0/1

entidades reales: “Torrejón de Ardoz”, “París”. Detectado: “París”. IoU: 1/2

entidades reales: “Torrejón de Ardoz”, “París”. Detectado: “París”, “hola”. IoU: 1/3

Librerías:

- a. <https://github.com/LIAAD/yake> : NER unsupervised
- b. <https://github.com/chartbeat-labs/textacy> : extension de spacy
- c. <https://github.com/boudinfl/pke> : keyphrase extraction

Pendiente Fran: chequear qué ocurre si se analiza el corpus de una tacada o conjuntos de documentos.

- Fran: librería que usa el NER de spacy y busca en base de conocimiento de wikipedia esa entidad, dándonos el link. En inglés y francés, pero español?  
<https://github.com/Lucaterre/spacyfishing#Use-other-language>

## Cosas a explorar

### NER

Objetivo:

- Chequear el grado de calidad de diferentes NER, mono y multilinguaje
- 
- Spacy: <https://spacy.io/models/xx> (multilinguaje)
- \*\* <https://spacy.io/models/es> (Pedro usa es\_core\_news\_lg, solo español)

Meterse paquete a paquete y chequear si tiene NER:

<https://spacy.io/usage/models>

- Huggingface:
- [https://huggingface.co/models?pipeline\\_tag=token-classification&sort=downloads](https://huggingface.co/models?pipeline_tag=token-classification&sort=downloads)
- <https://huggingface.co/Davlan/bert-base-multilingual-cased-ner-hrl>
- <https://huggingface.co/mrm8488/bert-spanish-cased-finetuned-ner>

Dataset NER: [https://huggingface.co/datasets/polyglot\\_ner](https://huggingface.co/datasets/polyglot_ner)

Lista de un montón de datasets NER: <https://metatext.io/datasets-list/ner-task>

Evaluar NERs en spacy: <https://spacy.io/api/cli#evaluate>

## Comparar diferentes métodos de clusterización:

- U-map
- LDA

## Comparar diferentes métodos de embedding:

- Word2vec
- Glove
-

- Topic2vec
- Paragraph2vec
- Diferentes transformers de huggingface. Guiarse por rankings de allí
- BM-25