

Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ingeniería de Transporte y Logística



# Sistemas Urbanos Inteligentes

Aprendizaje de embeddings

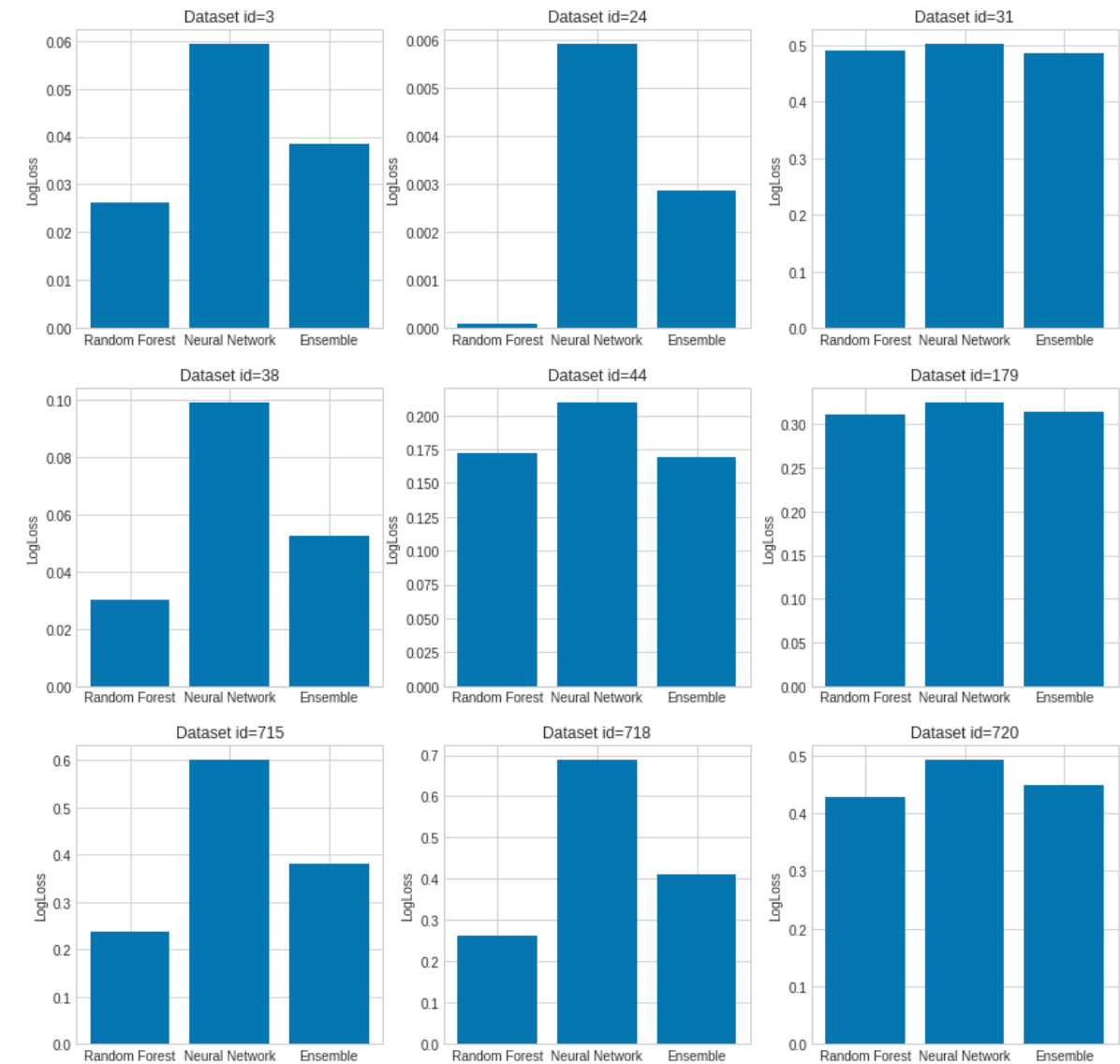
Hans Löbel

Dpto. Ingeniería de Transporte y Logística  
Dpto. Ciencia de la Computación

La aplicación típica de los MLP es en el aprendizaje sobre datos tabulados

- Filas en una tabla pueden verse de manera natural como los vectores de entrada.
- Pero parece que acá tampoco le va muy bien...

id	name	rows	cols
3	kr-vs-kp	3196	36
24	mushroom	8124	22
31	credit-g	1000	20
38	sick	3772	29
44	spambase	4601	57
179	adult	48842	14
715	fri_c3_1000_25	1000	25
718	fri_c4_1000_100	1000	100
720	abalone	4177	8



La aplicación típica de los MLP es en el aprendizaje sobre datos tabulados

- Filas en una tabla pueden verse de manera natural como los vectores de entrada.
- Sin embargo, usar un MLP “puro” tiende a generar rendimientos inferiores a métodos mucho más simples.
- El culpable es el teorema de aproximación universal (recuerden que podemos aprender todo, en particular, ruido en el set de entrenamiento).
- En general, es posible identificar 2 fuentes de aprendizaje espurio.

La aplicación típica de los MLP es en el aprendizaje sobre datos tabulados

## 1. Discordancia en tipos de dato

- Datos tabulados mezclan generalmente variables **categorías y numéricas**.
- Fuentes son **heterogéneas**, por lo que pueden tener **distintas unidades y escalas** de medición.
- Variables son dispersas (**sparse**), es decir, toman **pocos posibles valores**, incluso cuando son numéricas. Peor aún, puede haber **datos faltantes**.



Solución tradicional es a través de modelación (por ej. usando variables *dummies* y/o transformando variables)



La aplicación típica de los MLP es en el aprendizaje sobre datos tabulados

## 2. Distribución y correlación en los datos

- **Columnas** en datos tabulados presentan generalmente una **alta correlación**, por lo que basta un subconjunto pequeño de variables para poder hacer la predicción.
- Existe habitualmente un **alto desbalance** en las **clases** a predecir.



Solución tradicional es a través la ingeniería de datos/características y del muestreo

## Centrémonos inicialmente en el punto 1. Discordancia en tipos de dato

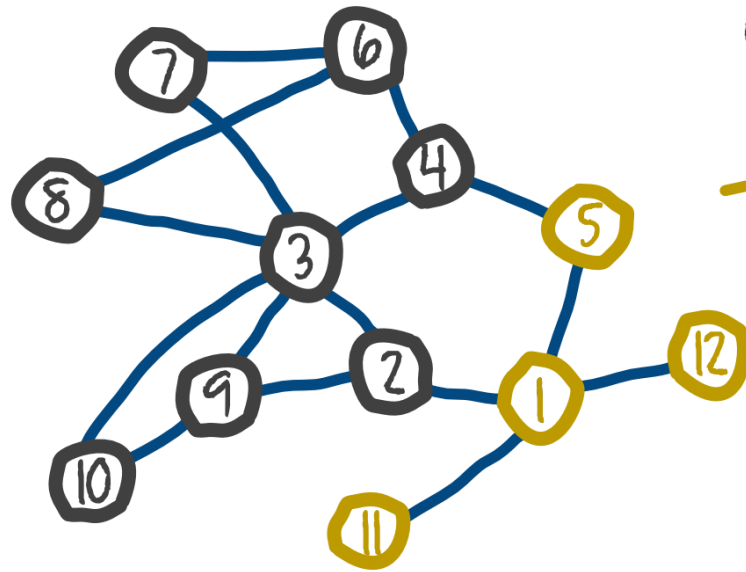
- Si bien la modelación es una herramienta fundamental, en este curso siempre tomaremos el enfoque del aprendizaje.
- Al analizar los problemas, notamos que “**todo se reduce**” a la incapacidad de las redes de trabajar con datos que no sean numéricos (continuos), como los categóricos.
- En vez de adaptar las redes para que funcionen con datos categóricos, buscaremos **aprender transformaciones** que muevan los datos categóricos a un espacio numérico.
- Esto tiene la ventaja que es posible capturar información semántica(?). Para hacer esto, utilizaremos el concepto de *embedding*.



¿Qué es eso de **embedding semántico**?

Un buen *embedding* captura información semántica, es decir, elementos que conceptualmente son cercanos, deben serlo también numéricamente.

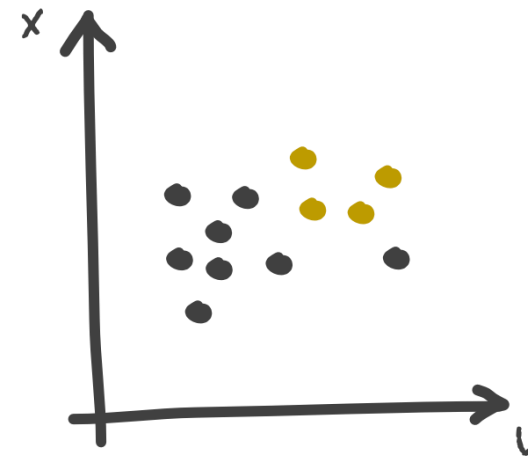
from a graph representation ...



embedding  
algorithm

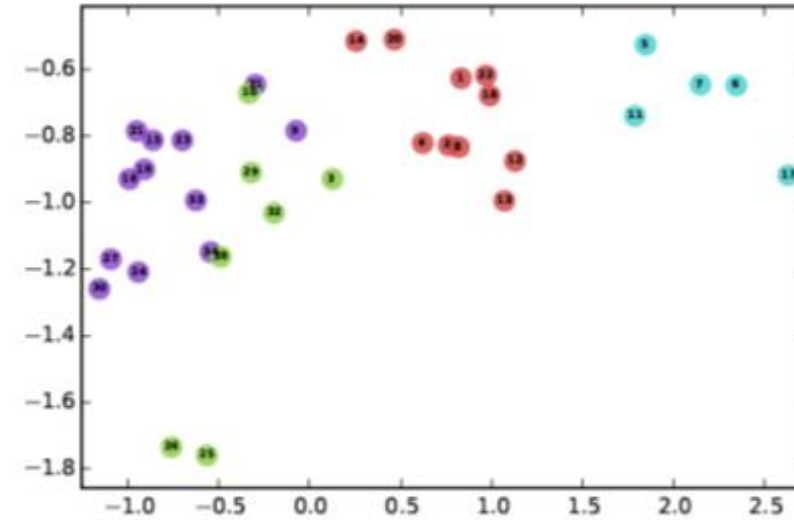


to real vector representation



¿Qué es eso de **embedding** semántico?

Un buen *embedding* captura información semántica, es decir, elementos que conceptualmente son cercanos, deben serlo también numéricamente.

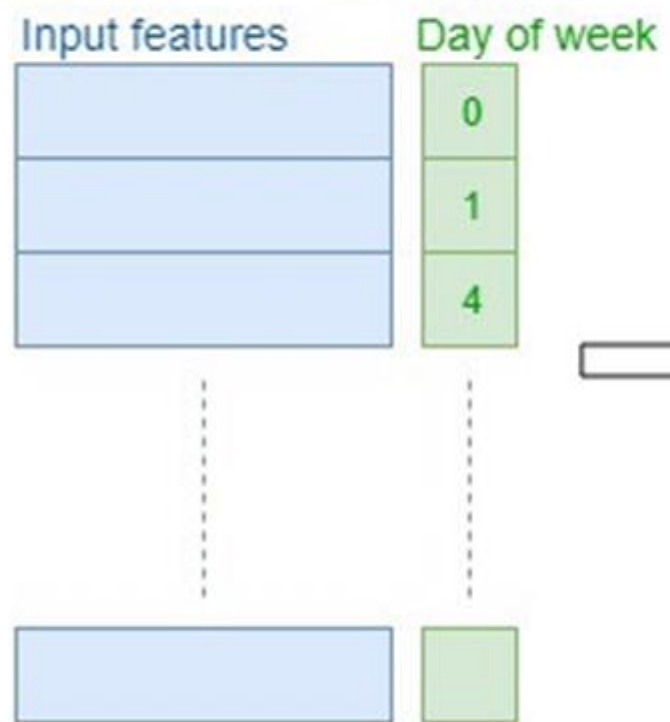




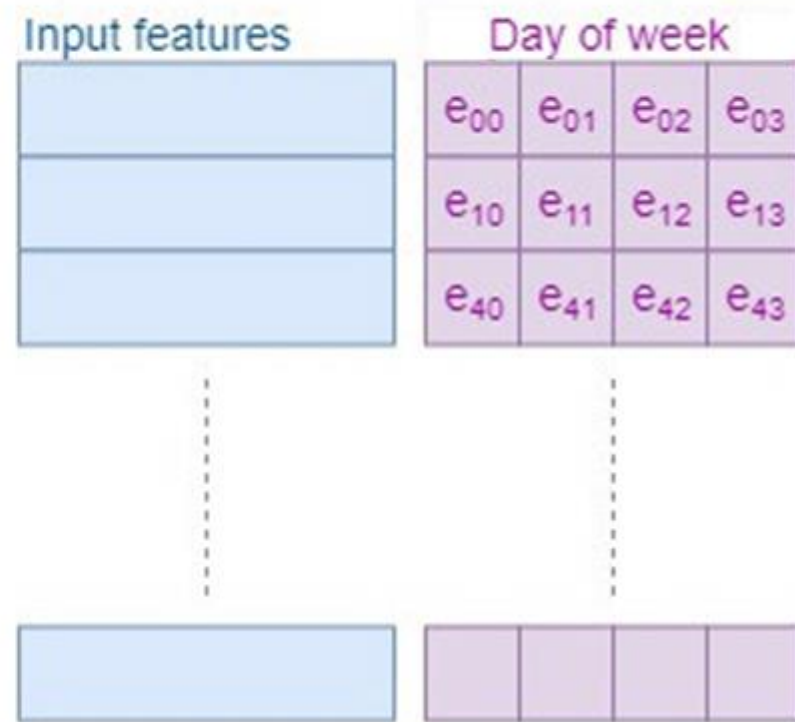
## ¿Qué es un *embedding*?

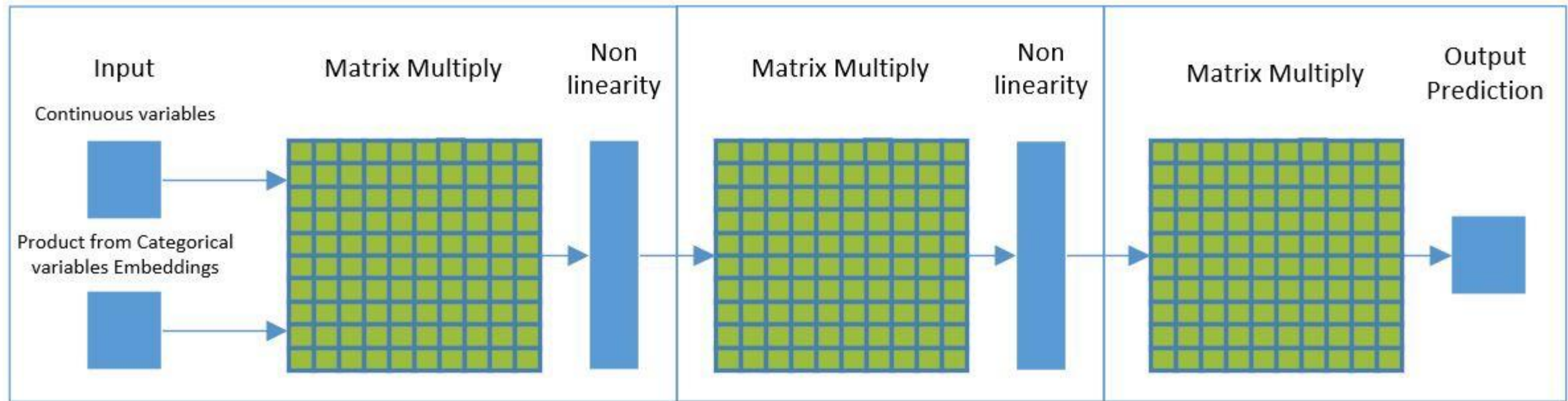
- En esencia, un *embedding* es un esquema para transformar datos que viven en espacios no numéricos, a un espacio vectorial continuo, generalmente de menor dimensionalidad (de ahí el nombre).
- En otras palabras, buscamos aprender una *representación* continua de datos no numéricos.
- Un *embedding* puede ser diseñado o aprendido para una tarea (o incluso preentrenado para otra tarea).
- En ML, no solo se usan para transformar datos categóricos, sino también para grafos, texto y otros.

## Input data

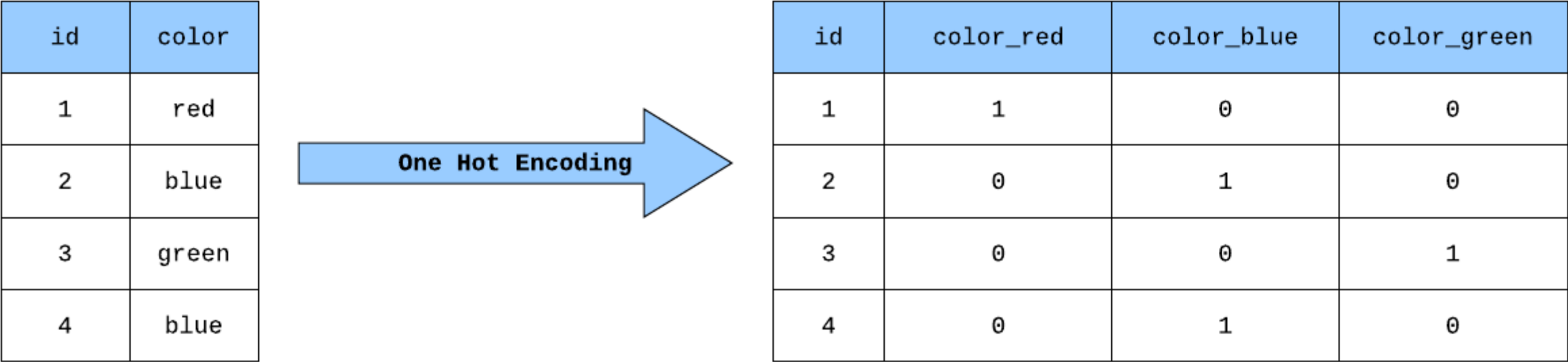


## Input data after embedding

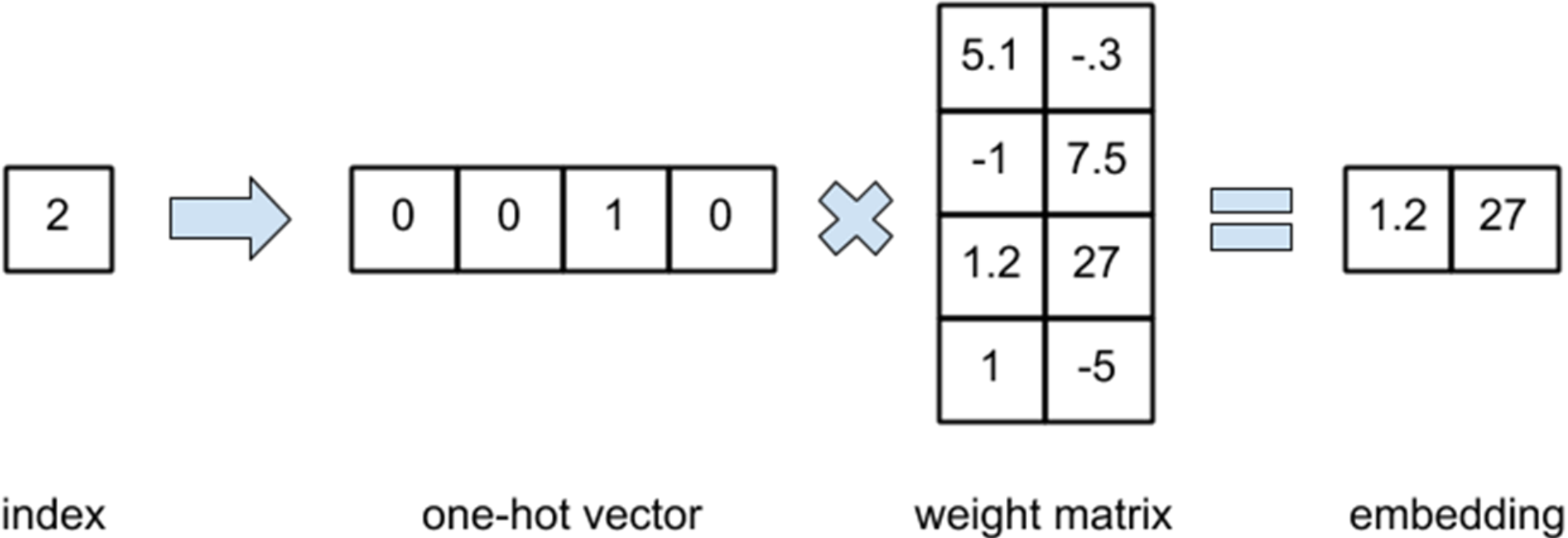




¿Como integramos este concepto de *embedding* en un MLP?

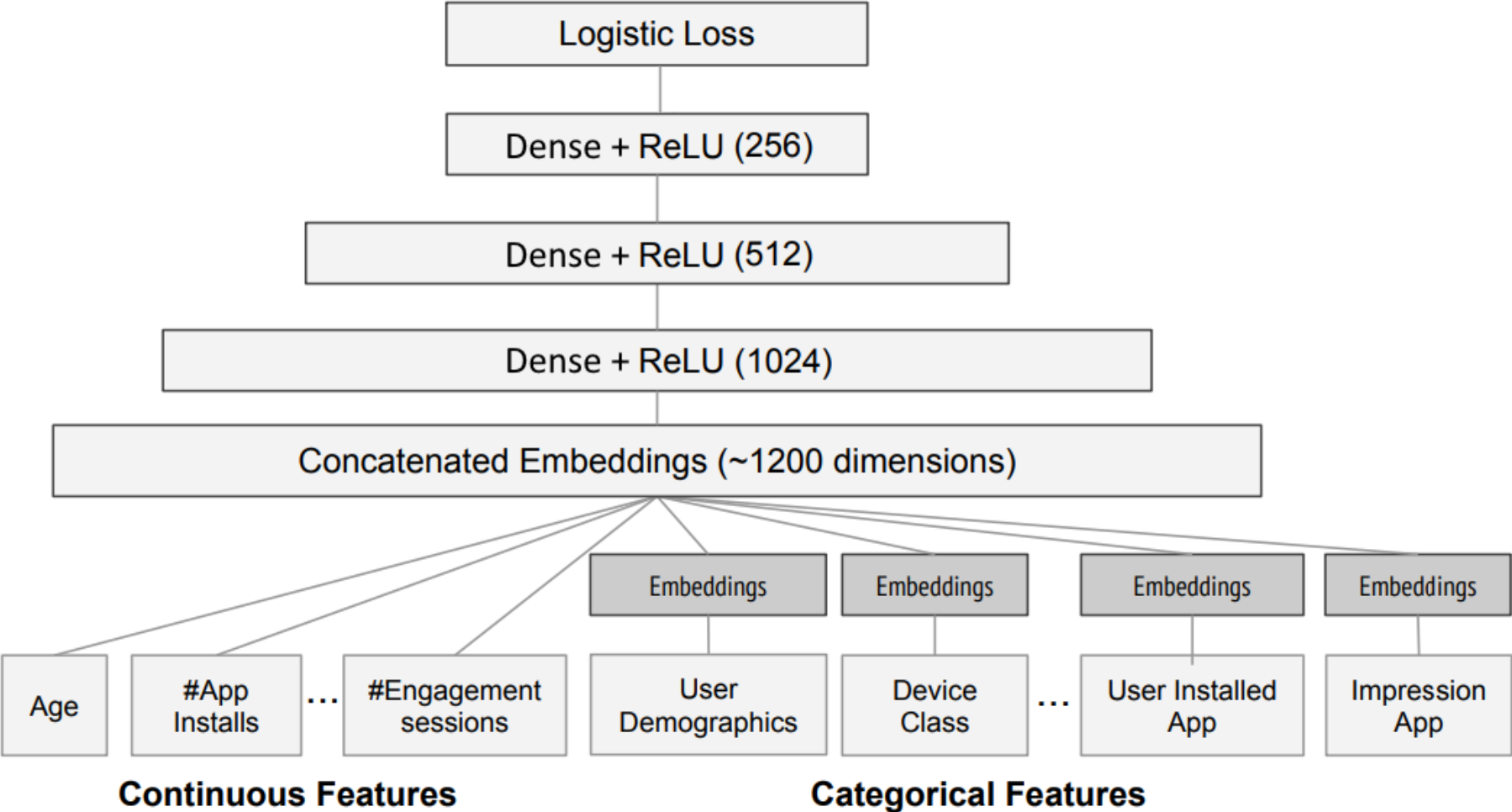


Al utilizar una codificación *one-hot* (*variables dummies*), es posible utilizar **productos matriciales** para generar un **vector asociado a cada categoría de la variable** (esencialmente, una indexación)





Finalmente, al tener productos matriciales involucrados, es posible empaquetarlo todo dentro de una red neuronal



Veamos ahora un caso de estudio

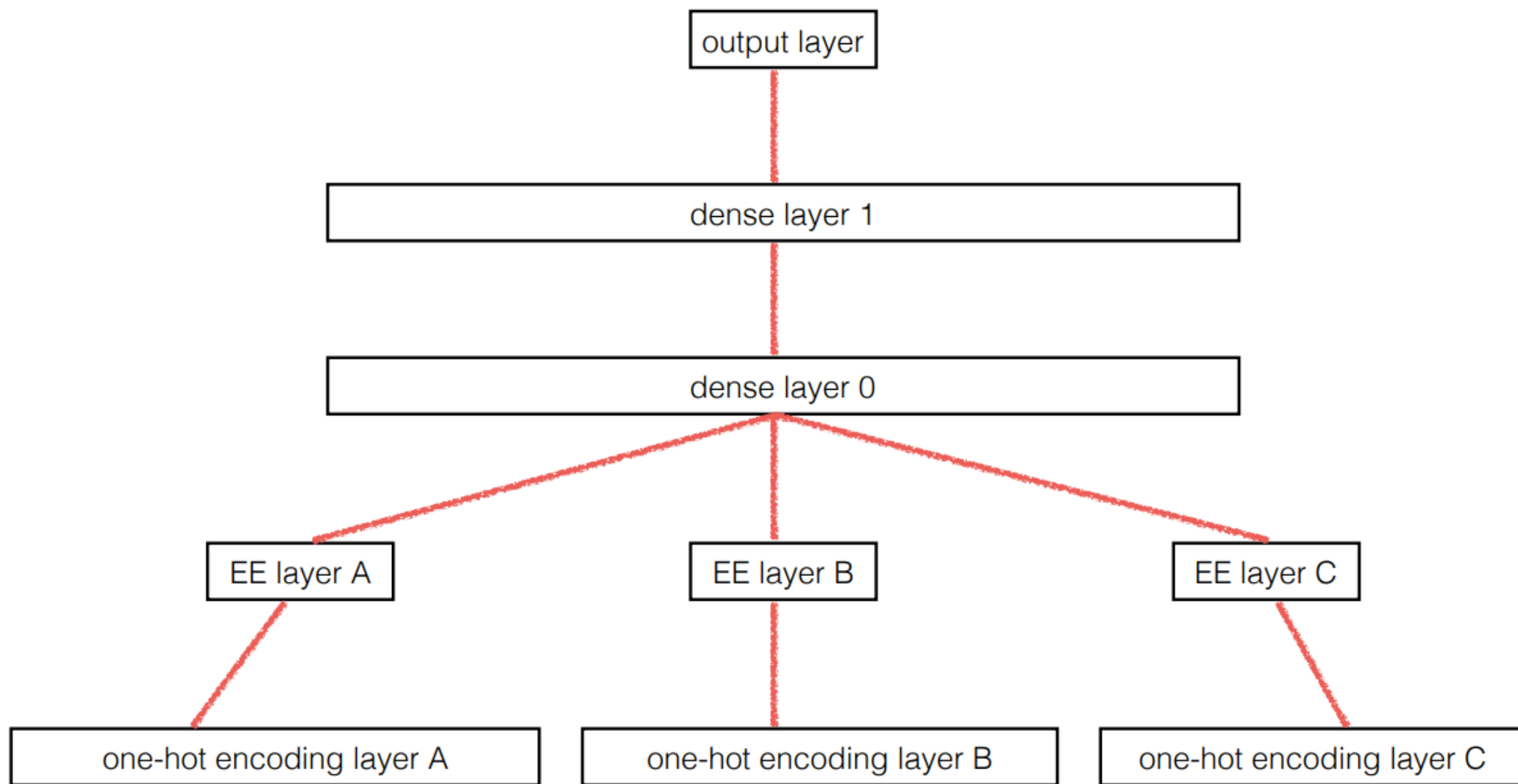
# Entity Embeddings of Categorical Variables

C. Guo, F. Berkhahn (2016)

Objetivo es predecir (log) ventas diarias de cada tienda de la cadena Rossman

- 2.5 años de datos para 1115 tiendas.
- 1.017.210 registros, divididos en 90% para entrenamiento y 10% para test.
- No se utilizó *feature engineering*.
- Compara el rendimiento de redes neuronales basadas en *embeddings* categóricos, vs otros modelos de ML: XGBoost, Random Forest, KNN.

feature	data type	number of values	EE dimension
store	nominal	1115	10
day of week	ordinal	7	6
day	ordinal	31	10
month	ordinal	12	6
year	ordinal	3 (2013-2015)	2
promotion	binary	2	1
state	nominal	12	6



$$MAPE = \left\langle \left| \frac{Sales - Sales_{predict}}{Sales} \right| \right\rangle$$

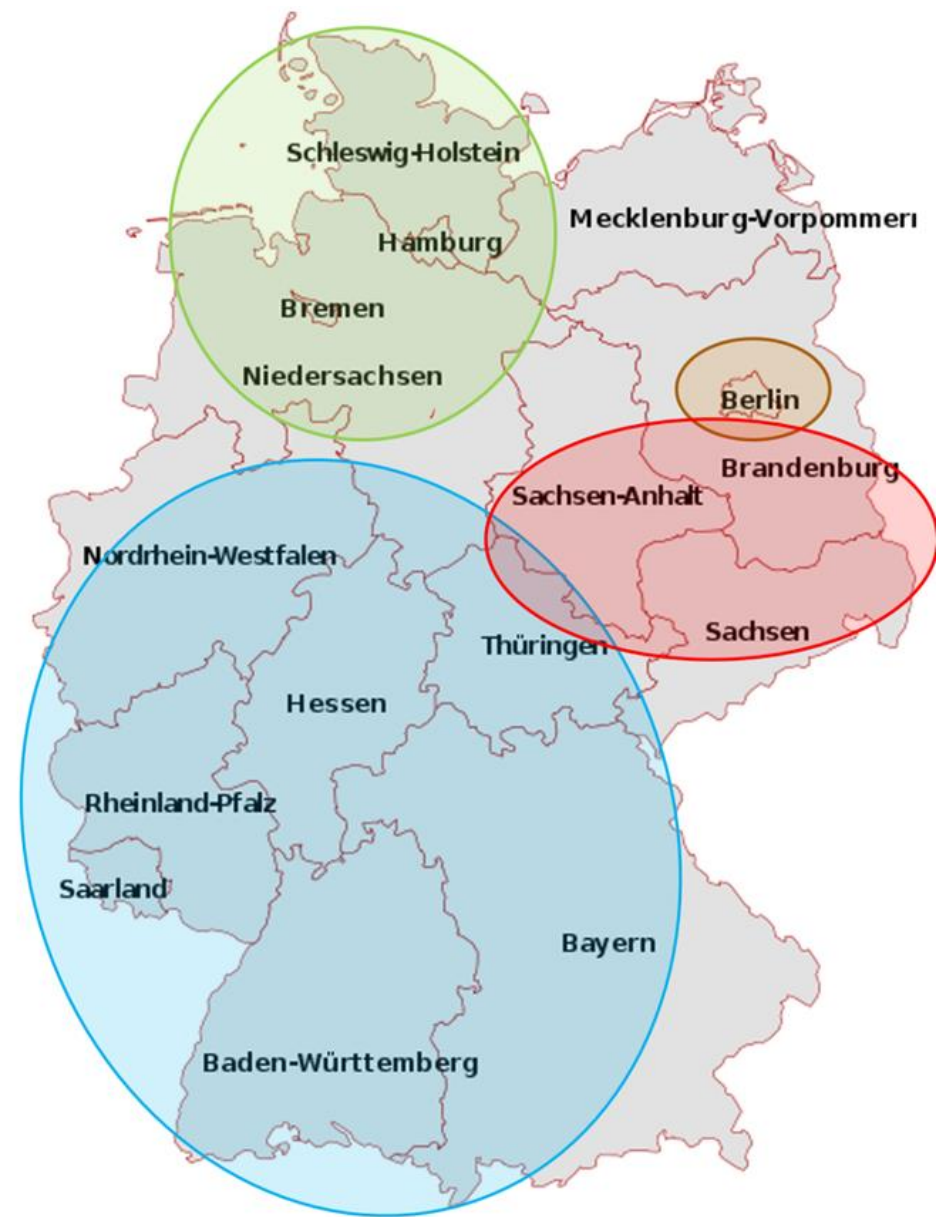
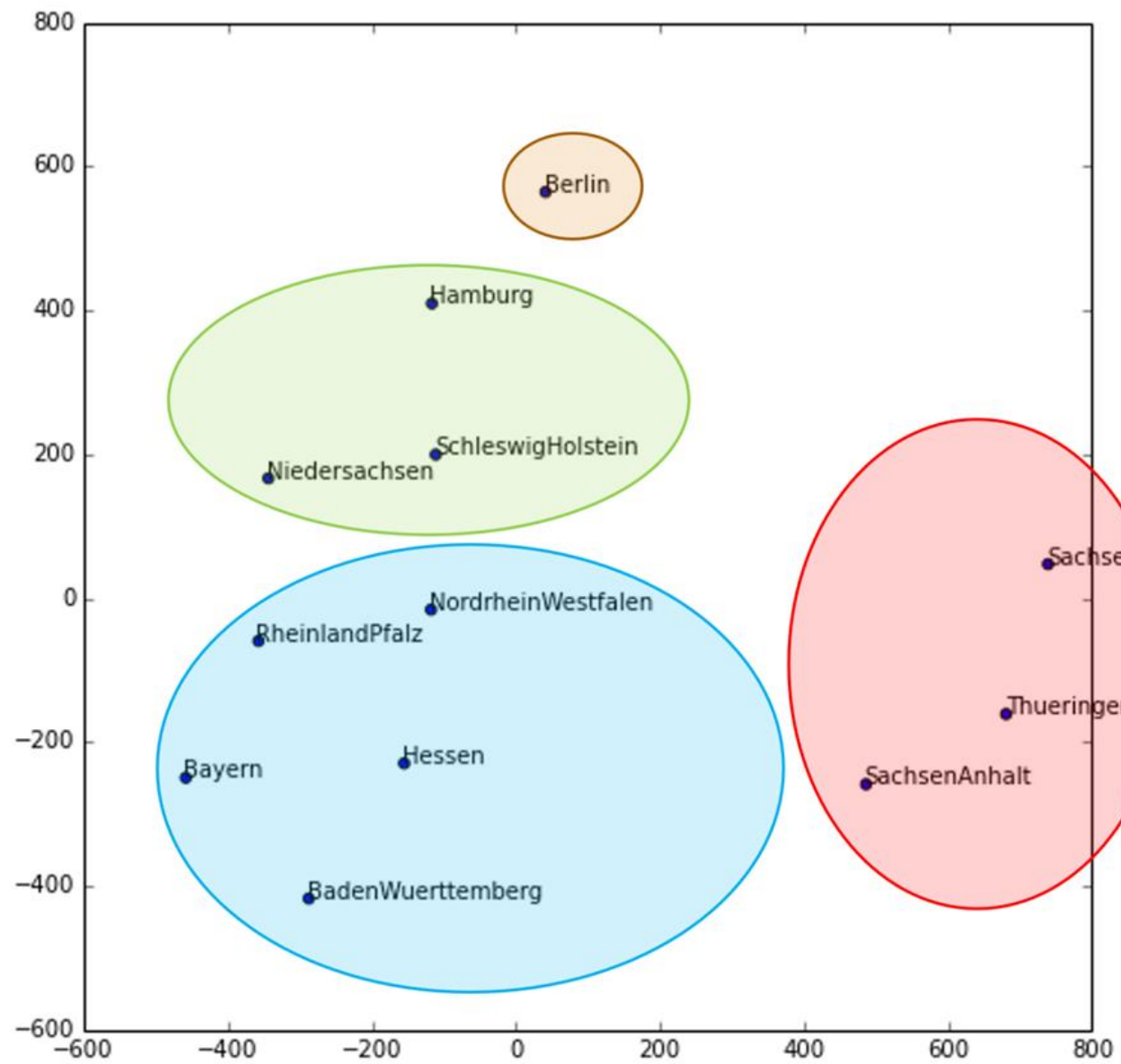
method	MAPE	MAPE (with EE)
KNN	0.315	0.099
random forest	0.167	0.089
gradient boosted trees	0.122	0.071
neural network	0.070	0.070

TABLE III. Comparison of different methods on the Kaggle Rossmann dataset with 10% shuffled data used for testing and 200,000 random samples from the remaining 90% for training.

method	MAPE	MAPE (with EE)
KNN	0.290	0.116
random forest	0.158	0.108
gradient boosted trees	0.152	0.115
neural network	0.101	0.093

TABLE IV. Same as Table IV except the data is not shuffled and the test data is the latest 10% of the data. This result shows the models generalization ability based on what they have learned from the training data.





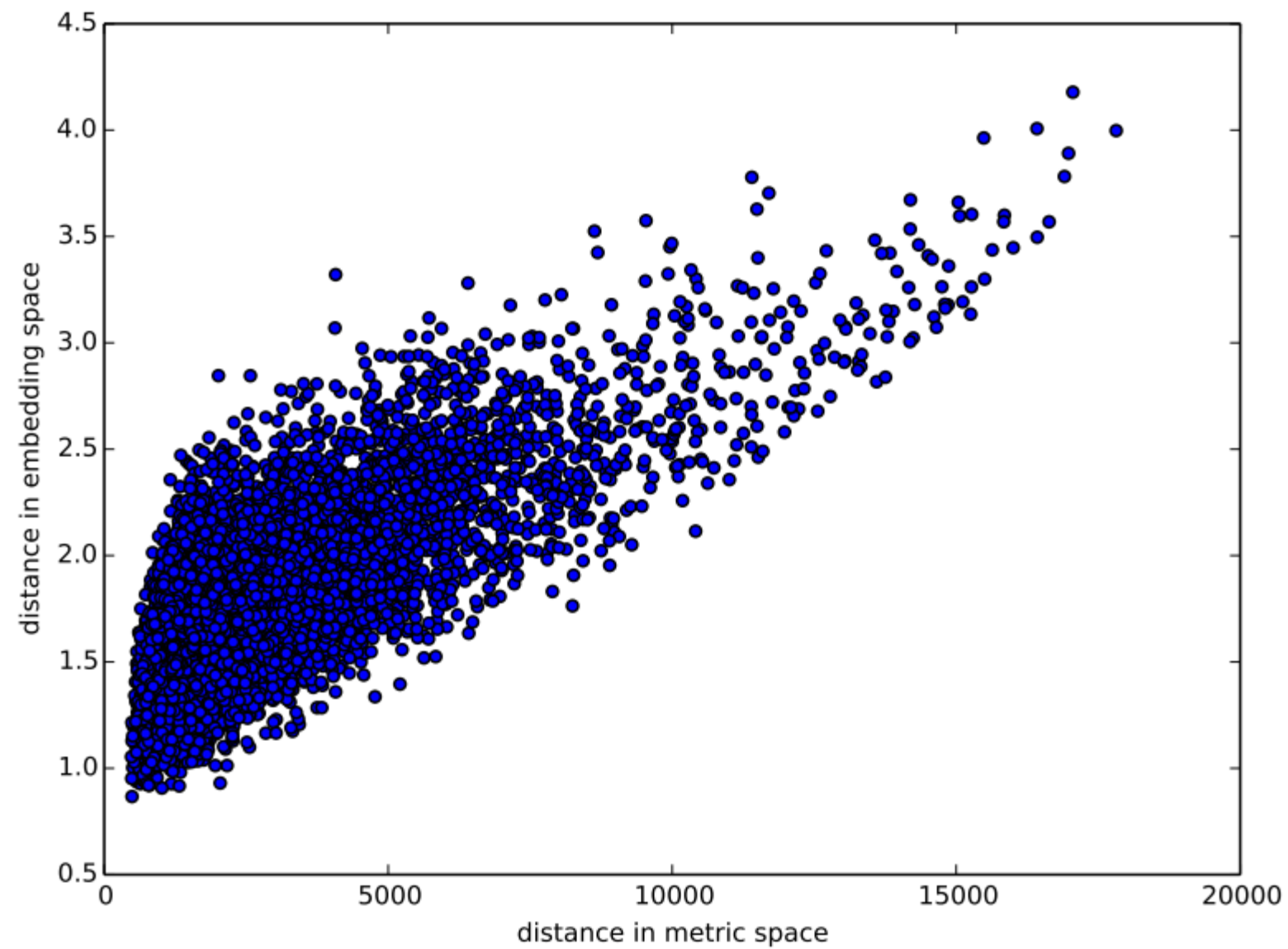
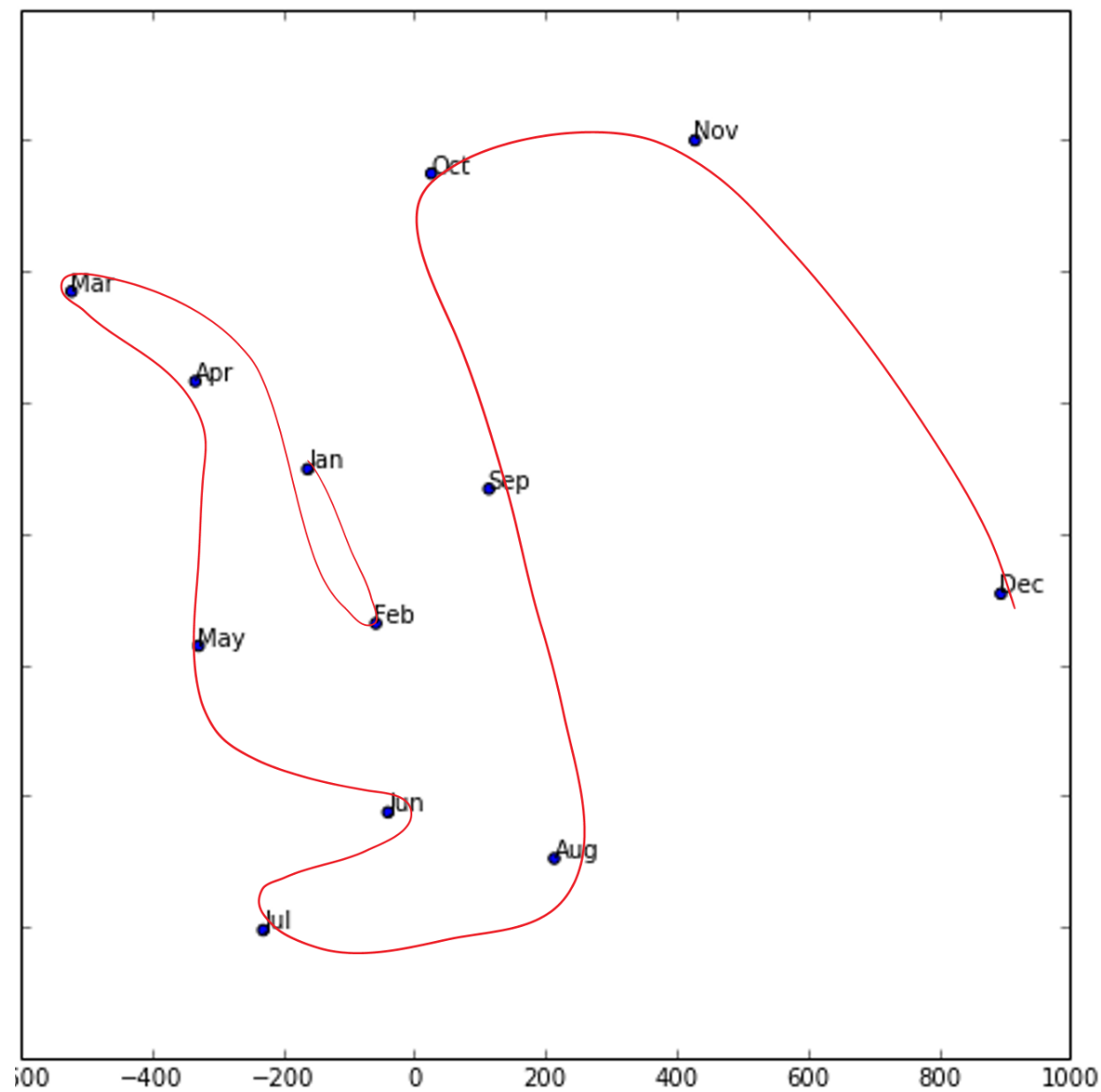
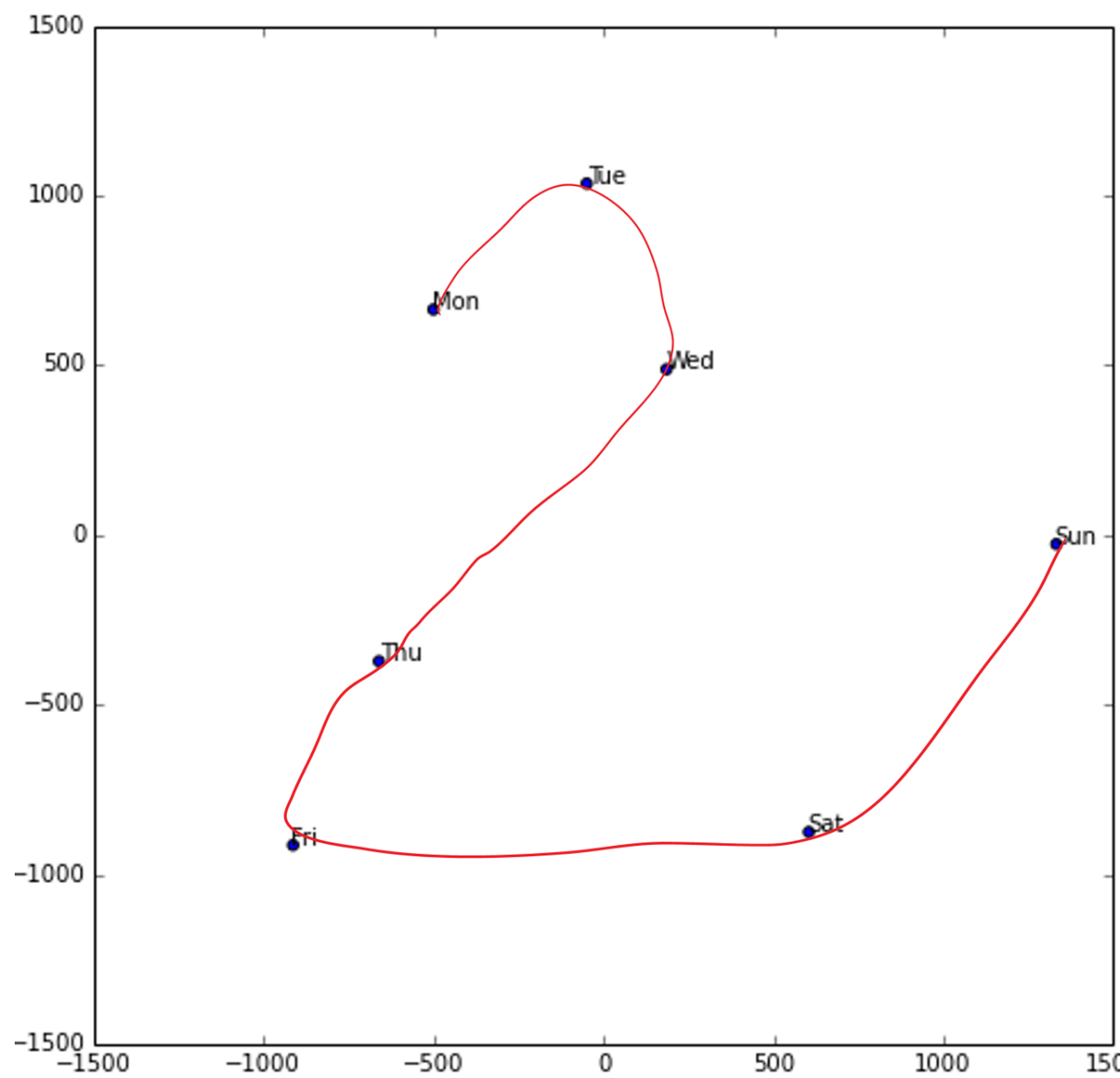


FIG. 2. Distance in the store embedding space versus distance in the metric space for 10000 random pair of stores.



## Sobre el punto 2. Distribución y correlación en los datos...

- Los problemas con la alta correlación entre variables y el alto desbalance de las clases son transversal a todas las técnicas de Deep Learning.
- Debido a esto, las técnicas para enfrentarlos las cubriremos en otros capítulos, al revisar los modelos donde originalmente fueron aplicadas (regularización, mecanismo de atención, y otros).
- Sin embargo, aunque hayan sido propuestas en otro contexto, siguen siendo aplicables a datos tabulados. Dos ejemplos son:
  - TabNet: Attentive Interpretable Tabular Learning (<https://arxiv.org/pdf/1908.07442.pdf>)
  - Neural Oblivious Decision Ensembles for Deep Learning on Tabular Data (<https://arxiv.org/abs/1909.06312>)

Vamos a Colab...





Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ingeniería de Transporte y Logística



# Sistemas Urbanos Inteligentes

Aprendizaje de embeddings

Hans Löbel

Dpto. Ingeniería de Transporte y Logística  
Dpto. Ciencia de la Computación