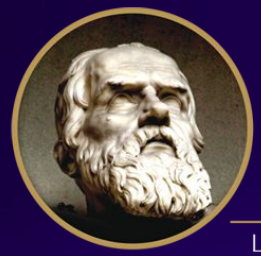
SPI '20

Introducción a Deep Learning

PRESENTACIÓN DE POSTERS



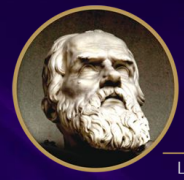
Galileo
UNIVERSIDAD
La Revolución en la Educación

SPI '20 Introducción a Deep Learning
PRESENTACIÓN DE POSTERS

Modelo de clasificación para predecir sobrevivencia después de un año de un ataque cardíaco

Luis Florian, René Tarot

https://github.com/lflorian/HeartAttack

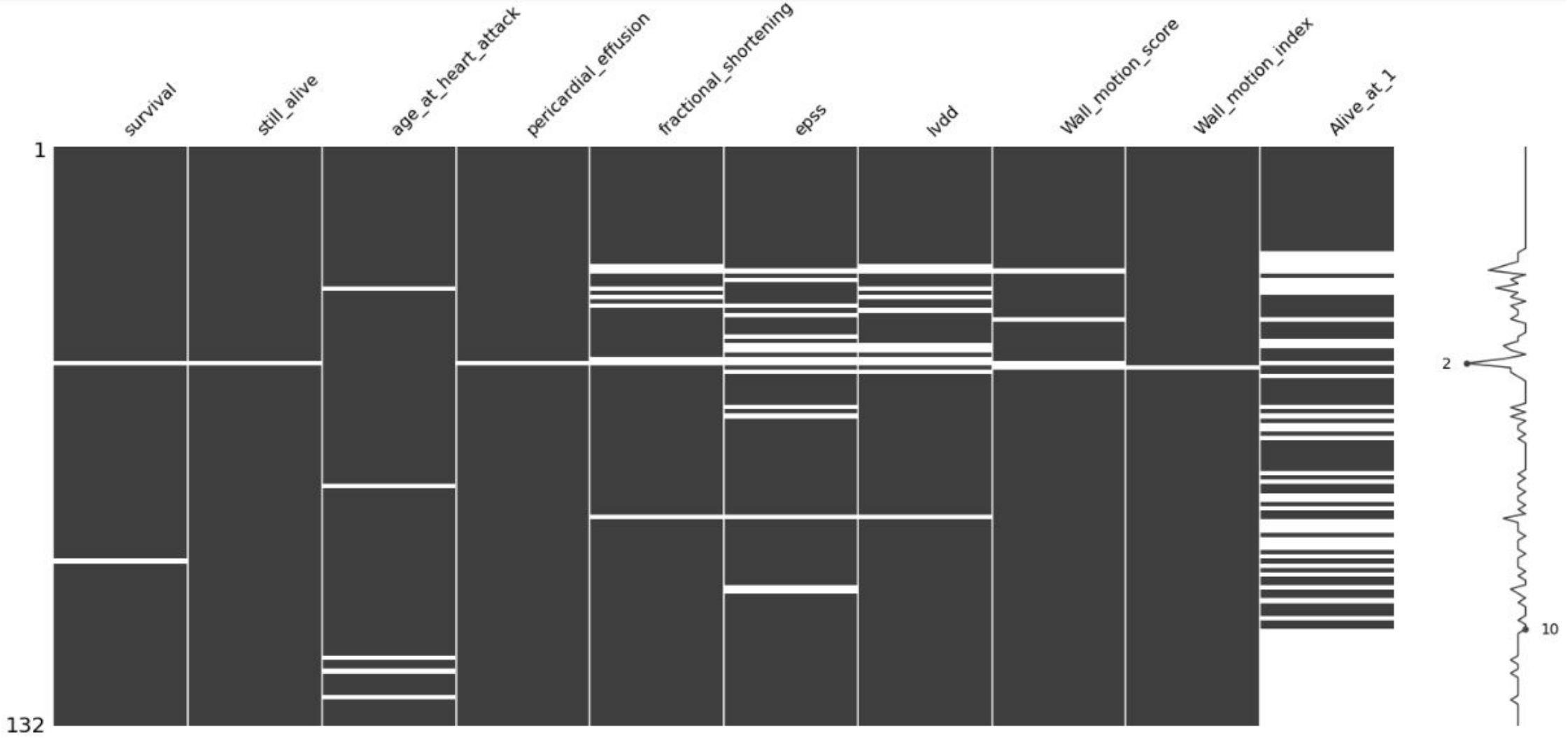


Galileo UNIVERSIDAD
La Revolución en la Educación

Exploring Dataset

survival	still-alive	age-at-heart-attack	...	name	group	Alive-at-1
11.0	0.0	71.0	...	name	1.0	0.0
19.0	0.0	72.0	...	name	1.0	0.0
16.0	0.0	55.0	...	name	1.0	0.0
57.0	0.0	60.0	...	name	1.0	0.0
19.0	1.0	57.0	...	name	1.0	0.0
...	...	...	...	...	...	...
7.5	1.0	64.0	...	name	NaN	NaN
41.0	0.0	64.0	...	name	NaN	NaN
36.0	0.0	69.0	...	name	NaN	NaN
22.0	0.0	57.0	...	name	NaN	NaN
20.0	0.0	62.0	...	name	NaN	NaN

Missing values

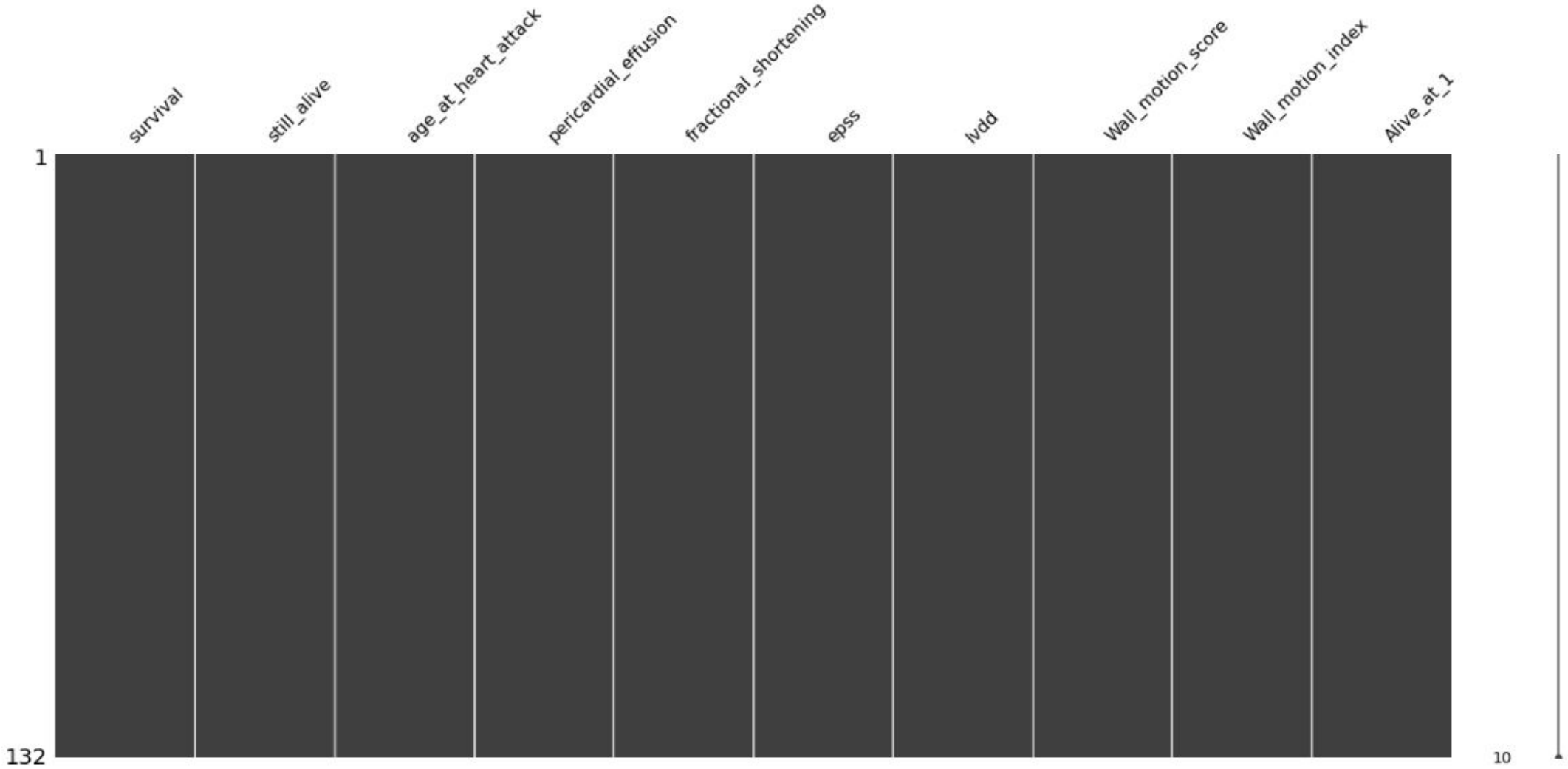


K-Nearest Neighbor Imputation

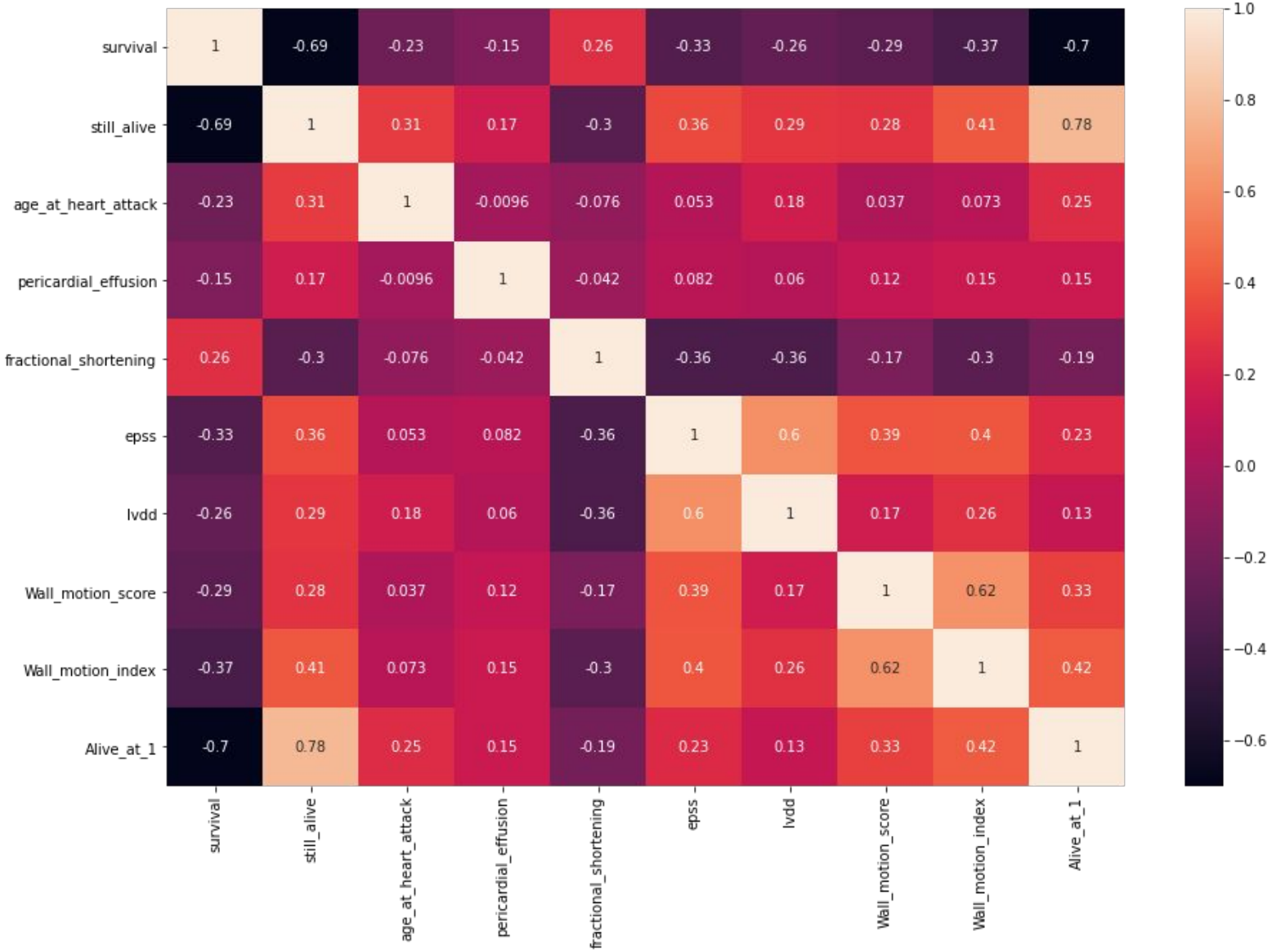
fancyimpute package

- Package contains advanced techniques
- Uses machine learning algorithms to impute missing values
- Uses other columns to predict the missing values and impute them
- Select K nearest or similar data points using all the non-missing features
- Take average of the selected data points to ll in the missing feature
- Nearest neighbor imputations which weights samples using the mean squared difference on features for which two rows both have observed data.

After Imputing



Correlation



Normalization

	survival	still_alive	age_at_heart_attack	pericardial_effusion	fractional_shortening	epss	lvdd	Wall_motion_score	Wall_m
0	0.192557	0.0	0.705882	0.0	0.416667	0.22500	0.511211	0.324324	
1	0.332982	0.0	0.725490	0.0	0.616667	0.15000	0.399103	0.324324	
2	0.280323	0.0	0.392157	0.0	0.416667	0.10000	0.246637	0.324324	
3	1.000000	0.0	0.490196	0.0	0.405000	0.30155	0.511883	0.378378	
4	0.332982	1.0	0.431373	0.0	0.250000	0.55000	0.769058	0.432432	
...	...	...	...	...	...	...	...	...	
127	0.131122	1.0	0.568627	0.0	0.383333	0.32250	0.538117	0.270270	
128	0.719150	0.0	0.568627	0.0	0.450000	0.13500	0.706278	0.243243	
129	0.631385	0.0	0.666667	0.0	0.316667	0.17500	0.612108	0.337838	
130	0.385642	0.0	0.431373	0.0	0.216667	0.40250	0.457399	0.351351	
131	0.350535	0.0	0.529412	0.0	0.233333	0.00000	0.491031	0.364865	

132 rows × 10 columns

Splitting Dataset

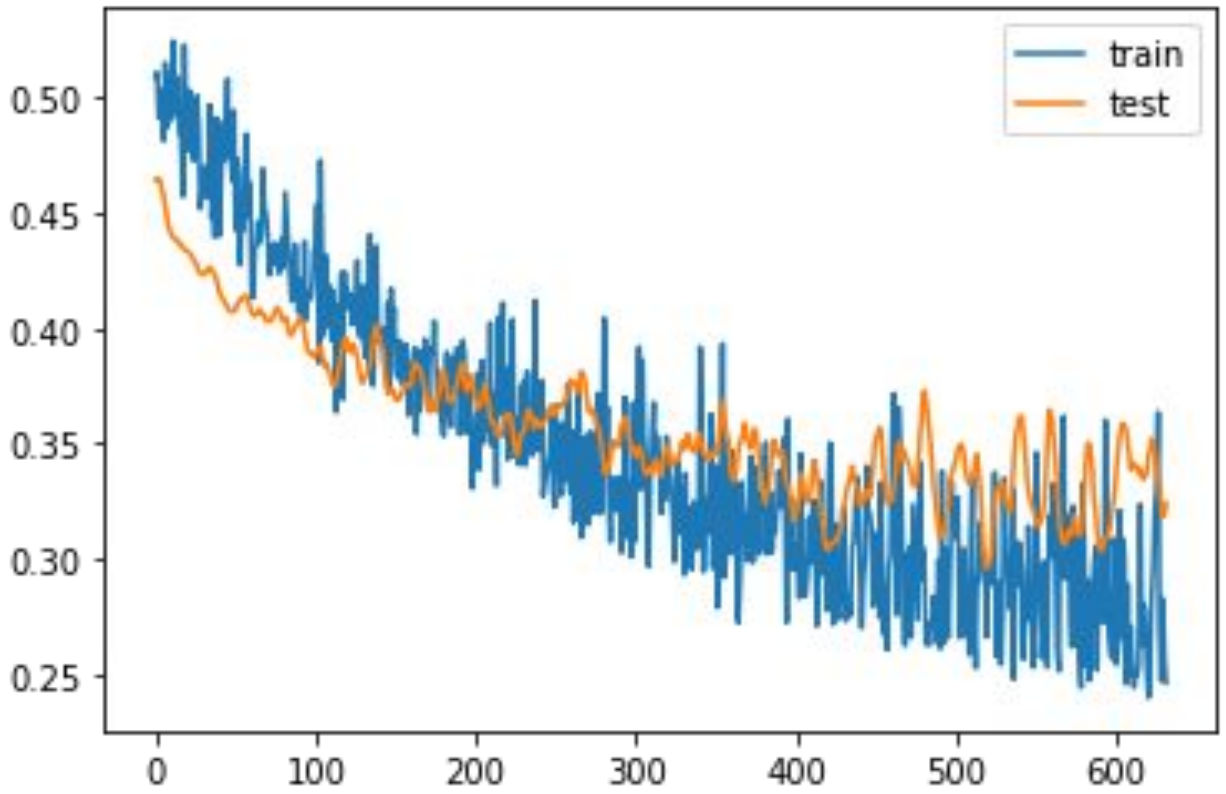
70% - Training

30% - Testing

Red Totalmente Conectada

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 8)                 80
_________________________________________________________________
dense_1 (Dense)              (None, 6)                 54
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 28
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 10
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 3
=================================================================
Total params: 175
Trainable params: 175
```

Resultados




```
Train: 0.882, Test: 0.900
```

Modelo de clasificación para predecir sobrevivencia después de un año de un ataque cardíaco

Luis Florian & René Tarot

Introducción

Este trabajo tiene como objeto mostrar los pasos que se llevaron a cabo durante la creación de un modelo de clasificación capaz de predecir lo más acertado posible si un paciente luego de subrir un ataque cardiacó sobrevivirá por lo menos un año.
El set de datos utilizado fue obtenido de una publicación de UCI Machine Learning Repository, la data fue recolectada por Dr. Evlin Kinney, The Reed Institute de Miami.
En este set de datos, todos los pacientes registrados sufrieron ataques cardiacos en algún momento del pasado. Algunos todavía están vivos y otros desafortunadamente no. Las variables survivel y still-alive, cuando se toman juntas, indican si un paciente sobrevivió durante al menos un año después del ataque cardiaco.
Utilizamos redes neurales para realizar dicho trabajo, y se obtuvo un resultado bastante aceptable.

Explorando el set de datos

survival	still-alive	age-at-heart-attack	...	name	group	alive-at-1
11.0	0.0	71.0	...	name	1.0	0.0
19.0	0.0	72.0	...	name	1.0	0.0
16.0	0.0	55.0	...	name	1.0	0.0
57.0	0.0	58.0	...	name	1.0	0.0
26.0	1.0	57.0	...	name	1.0	0.0
...	...	...	...	...	...	...
1.5	1.0	60.0	...	name	NaN	NaN
41.0	0.0	54.0	...	name	NaN	NaN
30.0	0.0	58.0	...	name	NaN	NaN
21.0	0.0	57.0	...	name	NaN	NaN
29.0	0.0	62.0	...	name	NaN	NaN

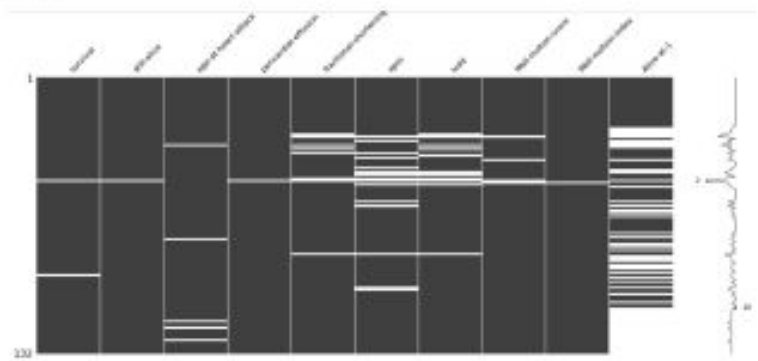
El set de datos consta de 132 instancias de pacientes y 12 variables categóricas y numéricas. En la publicación del Dataset se indica que las variables mult, name y group no aportan ningún valor. Por lo que obviarán en nuestro estudio.

Hipótesis

Aún siendo un dataset tan pequeño se podrá obtener un accurcy mayor a 80.

Valores Faltantes en el Dataset

Luego de revisar el dataset se detectaron 103 valores faltantes, los cuales es necesario cubrir debido al pequeño tamaño completo del dataset. Para esta tarea disponemos de varias opciones, sustituir por la media, la moda o utilizar una red neural para predecir los valores faltantes. Lo que después de pruebas nos dió un mejor resultados fue utilizar imputaciones, utilizamos Nearest neighbor imputation, el cual es una herramienta que pondera las muestras utilizando la diferencia de media cuadrática en features que en dos filas tienen data observada. Luego de su aplicación no quedó ningún valor faltante. A continuación se observa una gráfica que muestra la cantidad de valores faltantes representados por espacios en blanco.



Correlación entre variables

Para determinar que variables utilizar en nuestro modelo y cuales descartar utilizamos una correlación de variables. Los colores más claros representan una mayor correlación, al igual que un valor más cercano a 1.

Red totalmente conectada

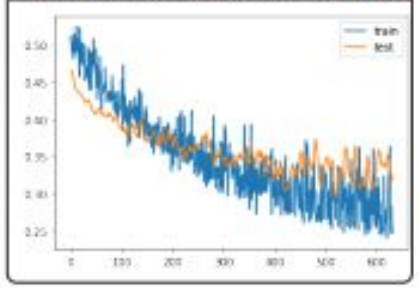
Se utilizó una red totalmente conectada de 5 capas, relu y sigmod como funciones de activación, la función adam de optimizer y binary crossentropy para función loss. Esta red Neural fue entrenada con 4000 epochs.

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 8)	88
dense_1 (Dense)	(None, 8)	72
dense_2 (Dense)	(None, 8)	72
dense_3 (Dense)	(None, 8)	72
dense_4 (Dense)	(None, 1)	9

Total params: 313
Trainable params: 313
Non-trainable params: 0

RESULTADOS: Train: 0.882, Test: 0.900

Gráfica de historial de Loss



Referencia

Calván, F. M. M. (s.f.). *Imputación de datos: teoría y práctica*.
Guttenberg, N. (2015). Learning to generate classifiers.
Kinney, D. E. (2002, month=March). *Echocardiogram data set*. Descargado de https://archive.ics.uci.edu/ml/datasets/echocardiogram