

ICU Mortality Prediction

Ana Sollars, Violeta Durán y Myriam Mateos

Procesado Masivo de Datos

Universidad Rey Juan Carlos

v.duran.2016@alumnos.urjc.es - a.sollars.2016@alumnos.urjc.es - m.mateos.2017@alumnos.urjc.es

(Actualizado el 29 de abril de 2020)

Grado en Ingeniería Biomédica

2 Tutorial

Challenge a resolver como base

Meredith Lee , Jesse Raffa , Marzyeh Ghassemi , Tom Pollard , Shweta Kalaridhi , Omar Badawi , Karen Matthys , Leo Anthony Celi

Published: Jan. 22, 2020, Version: 1.0.0

[WIDS \(Women in Data Science\) Challenge Announced! Register your team by 24 February 2020 \(Jan. 23, 2020\) 2:18 p.m.](#)

The WIDS Datalathon 2020 focuses on patient health through data from MIT's GCSOS (Global Open Source Severity of Illness Score) initiative. Brought to you by the Global WIDS team, the West Big Data Innovation Hub, and the WIDS Datalathon Committee. Winners will be announced at the WIDS Conference at Stanford University and via livestream, reaching a community of 100,000+ data enthusiasts across more than 50 countries.



When using this resource, please cite:

Lee, M., Haffa, J., Ghannem, M., Pollard, T., Kaleridhi, S., Bedoni, D., Matthys, K., Celi, L. A. (2022). WIDS (Women in Data Science) Datathon 2020: ICU Mortality Prediction (version 1.0.0). *PhysioNet*. <https://doi.org/10.13026/ucDe-th79>

Please include the standard citation for PhysioNet:

Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCH, Mark RG, Mietus JJ, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals (2003). *Circulation*. 107(23):e215-e220.

WIDS (Women in Data Science) Datathon 2020: ICU Mortality Prediction focuses on patient health through data from MIT's GCS5555 (Global Open Source Severity of Illness Score) initiative. Brought to you by the Global WIDS team, the West Big Data Innovation Hub, and the WIDS Datathon Committee, this year's datathon is launching on Kaggle: bit.ly/WIDSDatathon2020Kaggle.

The challenge is to create a model that uses data from the first 24 hours of intensive care to predict patient survival. MIT's GCSSES community initiative, with privacy certification from the Harvard Privacy Lab, has provided a dataset of more than 110,000 hospital intensive care unit (ICU) visits from patients, spanning a one-year timeframe. This data is part of a growing global effort and consortium spanning Argentina, Australia, New Zealand, Sri Lanka, Brazil, and more than 200 hospitals in the United States.

Training data are provided for model development; you will then upload your predictions for unlabeled test data to Kaggle and these predictions will be used to determine the leaderboard rankings.

Tutorials, sample code, other resources, and updates will be posted throughout the competition at wikiconference.org/dataset and on the Kaggle Discussion Forum. Winners will be announced at the WICOS Conference at Stanford University and via livestream, reaching a community of 100,000+ data enthusiasts across more than 70 countries.

[Contents](#)

Share



Access

Access Policy:

Anyone can access the files, as long as they conform to the terms of the specified license.

License (for files):

MIT License

Discovery

DOI:

<https://doi.org/10.1002/vech.201900079>

Topics:

icu mortality risk challenge
data science kaggle
predictive analytics
woman in data science

Project Website:

<https://bit.ly/W6125deletefrom20204escale>

Visualización de los Datos

Visualización del DataFrame

```
data.describe()
```

	encounter_id	patient_id	hospital_id	hospital_death	age	bmi	elective_surgery	height	icu_id	pre_icu_los_days	readmission
count	91713.000000	91713.000000	91713.000000	91713.000000	87485.000000	88284.000000	91713.000000	90379.000000	91713.000000	91713.000000	
mean	65606.079280	65537.131464	105.669262	0.086302	62.309516	29.185818	0.183736	169.641568	508.357692	0.835766	
std	37795.088538	37811.252183	62.854406	0.280811	16.775119	8.275142	0.387271	10.795378	228.989661	2.487756	
min	1.000000	1.000000	2.000000	0.000000	16.000000	14.844926	0.000000	137.200000	82.000000	-24.947222	
25%	32852.000000	32830.000000	47.000000	0.000000	52.000000	23.641975	0.000000	162.500000	369.000000	0.035417	
50%	65685.000000	65413.000000	109.000000	0.000000	65.000000	27.654655	0.000000	170.100000	504.000000	0.138889	
75%	98342.000000	98298.000000	161.000000	0.000000	75.000000	32.930206	0.000000	177.800000	679.000000	0.409028	
max	131051.000000	131051.000000	204.000000	1.000000	88.000000	67.814990	1.000000	195.590000	927.000000	159.090972	

8 rows x 178 columns

```
print('n° of variables : ', data.shape[1], '\nn° of patients: ', data.shape[0])
```

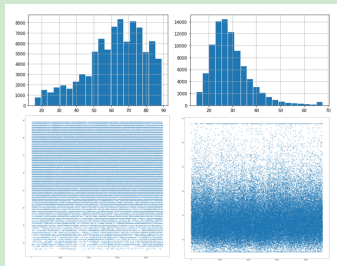
```

n# of variables : 186
n# of patients: 91713

```

Base de Datos original

- Datos estadísticos generales para ver la distribución de las variables.
- Elevado número de pacientes y variables.

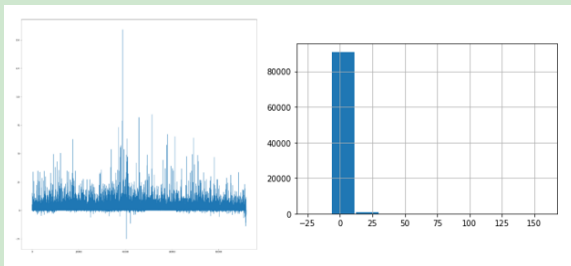


- Visualizamos los histogramas de cada variable
- Y los diagramas de dispersión.

- Estudiamos el significado clínico de las variables.
- Evaluamos si contienen valores erróneos, outliers y estudiamos la dispersión de los datos.
- Consideramos si es necesaria la transformación o no.

- Estudiamos el significado clínico de las variables.
- Evaluamos si contienen valores erróneos, outliers y estudiamos la dispersión de los datos.
- Consideramos si es necesaria la transformación o no.

Ejemplo



Según el diccionario proporcionado en el Challenge, obtenemos la siguiente descripción de la variable: "*The length of stay of the patient between hospital admission and unit admission.*"

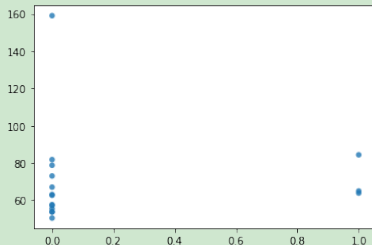
- Significado clínico: La duración del paciente en la UCI. La duración varía mucho dependiendo del tipo de paciente.
- Distribución: Parece una exponencial invertida, que decrece muy rápidamente
- **Importante:** Hay valores negativos, su única explicación puede ser un error humano. También destaca un único valor muy elevado.
- Los valores negativos vamos a tener que transformarlos. Mientras que los valores altos nos pueden interesar ya que pueden relacionarse con los *exitus*.

Ejemplo: Pre ICU los days

age	bmi	elective_surgery	ethnicity	gender	height	...	aids	cirrrosis	diabetes_mellitus	hepatic_failure	immunosuppression	leukemia	lymphoma
74.0	44.370252	0	Caucasian	F	149.90	...	0.0	0.0	1.0	0.0	1.0	0.0	0.0
73.0	NaN	0	Caucasian	M	167.60	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NaN	42.364193	0	Caucasian	M	167.60	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	27.887976	0	African American	M	175.30	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	33.448627	0	Caucasian	M	176.50	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...
NaN	30.470522	0	Caucasian	F	168.00	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	21.947625	0	Caucasian	F	157.48	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	24.445450	0	Caucasian	M	188.00	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	18.526701	0	Caucasian	M	178.00	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	33.933518	0	Caucasian	F	152.00	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN

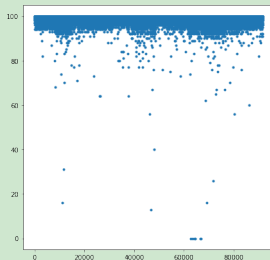
Vemos que es curioso que al visualizar únicamente los valores negativos, nos encontramos con muchos valores nulos. Pero al no poder determinar la razón exacta de esos valores nulos decidimos calcular el valor absoluto de los mismos. Se podría considerar que una *fecha vacía de salida* - *fecha de entrada* podría dar lugar a un valor negativo.

Ejemplo: Pre ICU los days



En este caso no vemos que haya relación entre la longitud de la estancia y la mortalidad. Por ello no modificamos estos valores.

Ejemplo



Según el diccionario proporcionado en el Challenge, obtenemos la siguiente descripción de la variable: *"The patient's highest peripheral oxygen saturation during the first 24 hours of their unit stay."*

Visualización de los Datos

Análisis de Datos

- Significado clínico: La saturación periférica en sangre máxima durante las primeras 24 horas en la UCI.
- Distribución: similar a una exponencial, que crece muy rápidamente según nos acercamos al 100 %.
- **Importante:** al ser pacientes ingresados en la UCI, es difícil saber cuando una saturación periférica baja es errónea o es un valor válido.
- Cuando esta variable es igual a 0, se entiende que es un NaN, ya que es incompatible con la vida.

Visualización de los Datos

Ejemplo: d1 oxi max

encounter_id	patient_id	hospital_id	hospital_death	age	bmi	elective_surgery	ethnicity	gender	height	hospital_admit_source	icu_admit_source	icu_id	
62790	82571	1570	43	0	81.0	49.481875	0	Caucasian	F	167.0	Emergency Department	Accident & Emergency	642
63628	3601	49594	43	0	57.0	36.285877	0	Caucasian	M	173.0	NaN	Accident & Emergency	642
64479	53429	36192	43	0	56.0	32.981302	0	Caucasian	F	152.0	Emergency Department	Accident & Emergency	642
64599	43351	99881	43	0	79.0	NaN	1	Caucasian	F	NaN	Recovery Room	Operating Room / Recovery	642
66590	15517	20799	43	0	NaN	26.543210	0	Caucasian	M	180.0	Emergency Department	Accident & Emergency	642
66859	32251	95313	43	0	71.0	34.915968	0	Caucasian	M	173.0	Emergency Department	Accident & Emergency	642

6 rows x 136 columns

Vemos que son pacientes de distintas edades y que están vivos. Por lo tanto que la saturación sea igual a 0 se debe a que es un NaN.

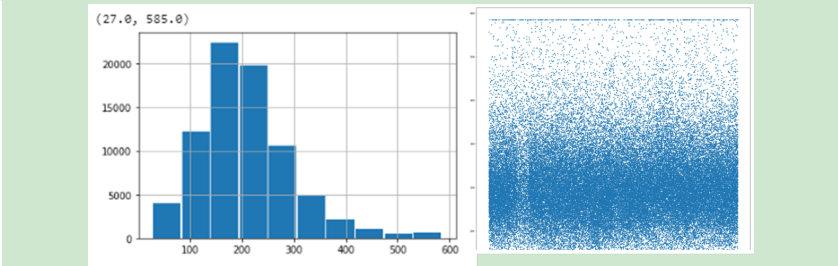
Visualización de los Datos

Ejemplo: d1 oxi max

psco2_apache	psco2_for_ph_apache	pao2_apache	ph_apache	...	h1_inr_max	h1_inr_min	h1_lactate_max	h1_lactate_min	h1_platelets_max	h1_platelets_min	h1_potassium_max
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Se trata de solo 6 pacientes con muchos missing datas, por lo que se decide eliminarlos de la base de datos.

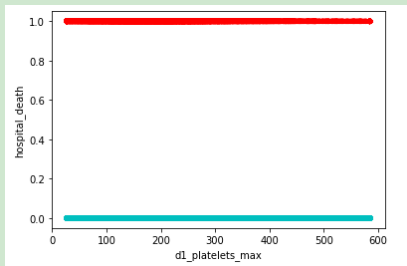
Ejemplo



d1 platelets max

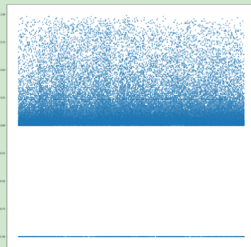
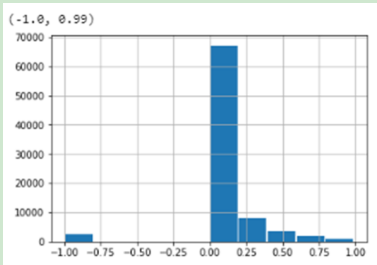
Según el diccionario proporcionado en el Challenge, obtenemos la siguiente descripción de la variable: *"The highest platelet count for the patient during the first 24 hours of their unit stay."*

- Significado clínico: La cantidad normal de plaquetas en la sangre es de 150 a 400 109/L.
- Distribución: similar a una normal, con valores centrados en torno a los 200uds.
- **Importante:** hay muchos pacientes con niveles de plaquetas bajos. Se puede deber a sangrados o anemia.
- En este caso, los valores extremos nos interesan, ya que pueden estar relacionados con los *exitus*.



Visualización de los Datos

Ejemplo



apache 4a hospital death prob

Según el diccionario proporcionado en el Challenge, obtenemos la siguiente descripción de la variable: *"The APACHE IVa probabilistic prediction of in-hospital mortality for the patient which utilizes the APACHE III score and other covariates, including diagnosis."*

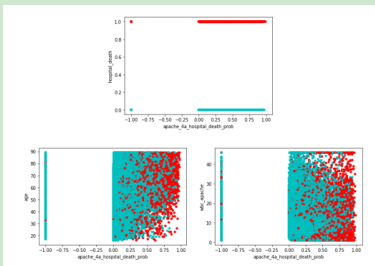
Visualización de los Datos

Análisis de Datos

- Significado clínico: probabilidad del riesgo de mortalidad según clasificaciones Apache.
- Distribución: similar a una distribución exponencial.
- **Importante:** existen valores negativos (**2371 casos**), las probabilidades deben tener un rango entre $[0,1]$.
- En este caso, se considera emplear la normalización logarítmica y se evalúa la transformación de los valores erróneos (negativos: -1).

Visualización de los Datos

Ejemplo: apache 4a hospital death prob



- A mayor valor de apache, mayor probabilidad de muerte por lo general.
- Hay pocos casos de valores negativos que corresponden en otras variables corresponden con *NaN*. Ej: Máximo 780 casos comparado con 'wbc apache'.
- La variable tiene 7947 casos de *NaN* de los 91713 pacientes totales.
- 203 casos de valores negativos que corresponden a *exitus*. (Importancia de imputar valores).
- Decidimos reemplazar los valores negativos por *NaN* e imputar la moda.

Preprocesado

1. Eliminar variables con muchos valores perdidos

Vemos el número de valores perdidos para cada variables. Establecemos un umbral, y en el caso de que éste sea muy elevado, eliminamos la variable ya que no nos aportará suficiente información. Pasamos de 186 variables originales a tan sólo 112, un número mucho más manejable.

```
threshold = len(data) * 0.60
df_thresh=data.dropna(axis=1, thresh=threshold)
df_thresh.shape

(91713, 112)
```

2. Eliminar variables con valores únicos

Las variables con valores únicos corresponden a códigos para identificar a los pacientes en distintos escenarios. Debido a que no aportan información para la clasificación, además es recomendable no tener en cuenta dichas variables para ahorrar memoria y aumentar la velocidad de computo.

```
print(df_thresh.shape)
display(df_thresh.nunique())

(91713, 112)
encounter_id          91713
patient_id            91713
hospital_id           147
hospital_death         2
age                   74
...
leukemia                2
lymphoma                 2
solid_tumor_with_metastasis  2
apache_3j_bodysystem    11
apache_2_bodysystem     10
Length: 112, dtype: int64

df_thresh.drop(['encounter_id', 'patient_id', 'hospital_id', 'icu_id'], axis='columns', inplace=True)
```

Preprocesado

3. Tratar los valores nulos

A la hora de trabajar con los valores nulos vamos a dividir la base de datos en variables categóricas y variables numéricas.

- **Variables categóricas:** Imputamos los valores nulos por el valor mas frecuente (moda).
- **Variables numéricas:** Imputamos los valores nulos por la mediana.

4.Tratar los outliers

A la hora de preprocesar los outliers, hemos decidido analizar las variables de una en una. De esta manera vemos si el outlier es un error, por lo que hay que imputarlo; o si es un valor posible pero atípico, y por ello lo tenemos que mantener.

Preprocesado

5. One Hot Encoding

Utilizamos esta técnica para codificar las variables categóricas. De forma que transformamos las categorías en valores numéricos.
Se realiza para 8 variables categóricas.

	gender_F	gender_M
0	0	1
1	1	0
2	1	0
3	1	0
4	0	1
...
91708	0	1
91709	1	0
91710	0	1
91711	1	0
91712	1	0

Clasificación

1. Dividir en train y test

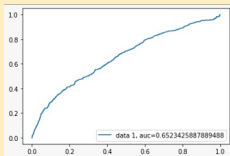
Una vez tenemos los datos preprocesados, pasamos a dividirlos en dos subconjuntos: train y test. Lo hacemos de forma aleatoria.

2. Normalización

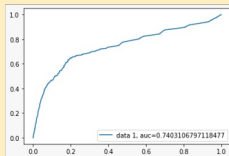
Probamos los siguientes tipos de normalización:

- Estandarización.
- Normalización Min-Max.

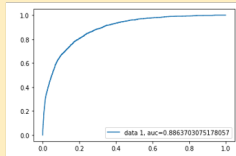
Min-max



Estandarización



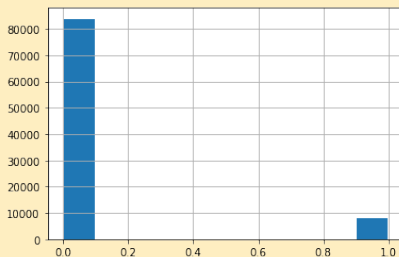
Sin normalizar



Debido a que los resultados según las curvas ROC no mejoran empleando la normalización, tendremos en cuenta las prestaciones sin normalización.

Clasificación

3. Balancear los datos



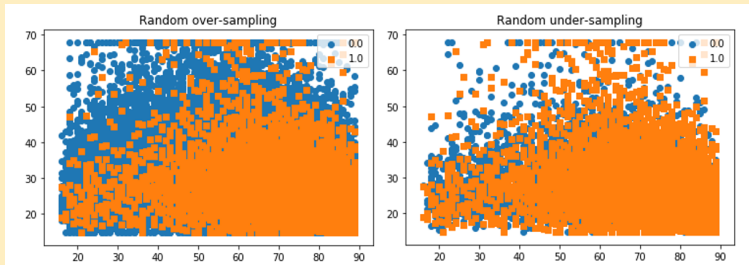
Como observamos nuestros datos están completamente desbalanceados (respecto a la salida). Esto implica que a la hora de diseñar un modelo de clasificación, el sistema no va a generalizar correctamente, ya que tiene mucho más peso los casos *no-exitus*.

Clasificación

3. Balancear los datos

Hemos seguido 2 estrategias diferentes (sobre los datos de training):

- **Undersampling**: Se eliminan ejemplos de la clase mayoritaria.
- **Oversampling**: Se duplican ejemplos de la clase minoritaria.



Clasificación

4. Clasificadores

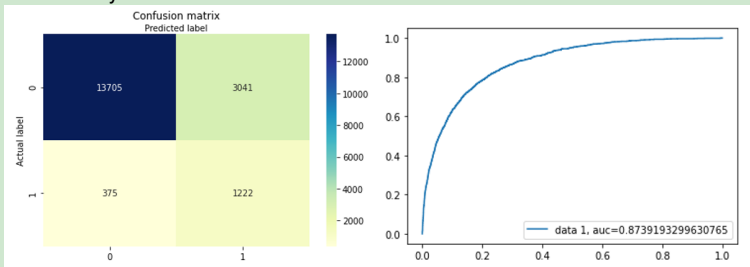
Hemos usado diferentes clasificadores tanto lineales como no lineales:

- **LDA:** *Linear discriminant analysis*, es una técnica de reducción de dimensionalidad.
- **Mayoría de votos:** Se utilizan varios clasificadores, la clasificación resultante corresponde a la clase que más votos ha recibido o predicciones.
- **Boosted trees:** Se basan en la combinación de modelos predictivos débiles (normalmente árboles de decisión) para crear un modelo predictivo fuerte. La generación de los árboles de decisión débiles se realiza de forma secuencial, creándose cada árbol de forma que corrija los errores del árbol anterior. Uno de los parámetros de este tipo de argumentos es la *tasa de aprendizaje*, que controla el grado de mejora de un árbol respecto del anterior.
- **Random forest:** Algoritmo de clasificación basado en muchos árboles de decisión.
- **SVM:** Una máquina de vectores de soporte construye un hiperplano óptimo en forma de superficie de decisión, de modo que el margen de separación entre las dos clases en los datos se amplía al máximo. Los vectores de soporte son un pequeño subconjunto de las observaciones de entrenamiento que se utilizan como soporte para la ubicación óptima de la frontera de decisión.

Clasificación

Ejemplo

A la hora de analizar los resultados del clasificador hemos tenido en cuenta la **matriz de confusión** y la **curva ROC**.



- Calculamos medidas de prestación como la **sensibilidad** y el **accuracy** a partir de la matriz de confusión.
- También tenemos en cuenta el **AUC**, que es el área bajo la curva ROC.

Clasificación

5. Comparación clasificadores

Para ver qué modelo de clasificación ha obtenido mejores resultados hacemos una tabla comparando las medidas de prestación de cada uno.

Oversampling

	LDA	Mayoria de votos	Boosted Trees	Random Forest
Accuracy	0.816933	0.897836	0.813389	0.923895
AUC	0.875789	0.885272	0.884251	0.879907
Sensibilidad	0.768942	0.594865	0.787101	0.282405

Undersampling

	LDA	Mayoria de votos	Boosted Trees	Random Forest
Accuracy	0.811372	0.796816	0.797198	0.788857
AUC	0.874223	0.884593	0.883193	0.879506
Sensibilidad	0.768316	0.810269	0.797746	0.810269

Unbalanced

	LDA	Mayoria de votos	Boosted Trees	Random Forest
Accuracy	0.918988	0.929237	0.92842	0.927602
AUC	0.866805	0.886996	0.883574	0.879692
Sensibilidad	0.406387	0.30119	0.313713	0.268003

Optimización de resultados

6. Selección de características

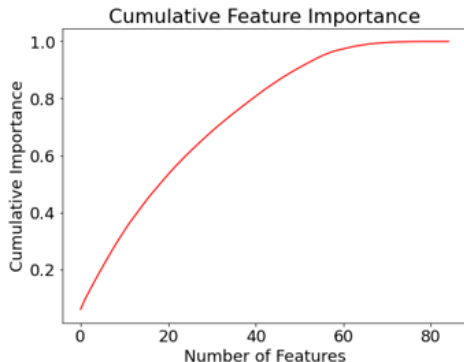
Vamos a intentar mejorar las medidas de prestación reduciendo el número de variables de los clasificadores, para ello:

- Vemos la importancia de cada variable.
- Algoritmo univariante.
- Eliminación recursiva de atributos.

Optimización de resultados

6. Selección de características

Feature importance: Una vez conocida la importancia de cada característica, podemos ver con qué numero de características obtenemos una *importancia acumulativa de 0,9*. De esta manera somos capaces de disminuir el numero de características:



50 features required for 0.90 of cumulative importance

Optimización de resultados

6. Selección de características

Feature importance: Resultados en clasificación.

	Normal	Reduction
Boosted trees		
Accuracy	0.8099002344218503	0.806247614893965
Sensibilidad	0.7896055103318722	0.7952410770194114
Mayoría de votos		
Accuracy	0.803412746006651	0.8944556506569263
Sensibilidad	0.802128991859737	0.5898559799624296

Optimización de resultados

6. Selección de características

Prueba F de Fisher: Es un algoritmo univariante, en el que se aplica una medida estadística para asignar una puntuación a cada atributo y luego se clasifican de acuerdo a su puntuación.

	Normal	Reduction
Boosted trees		
Accuracy	0.8099002344218503	0.8051572807065366
Sensibilidad	0.7896055103318722	0.7927363807138385
Mayoría de votos		
Accuracy	0.803412746006651	0.7947445892165949
Sensibilidad	0.802128991859737	0.802128991859737

Optimización de resultados

6. Selección de características

Eliminación Recursiva de atributos: Funciona seleccionando un conjunto de atributos como un problema de búsqueda, en donde las diferentes combinaciones son evaluadas y comparadas. Utilizamos un modelo predictivo, **ExtraTreesClassifier** y luego asignamos una puntuación a cada combinación basada en los resultados.

	Normal	Reduction
Boosted trees		
Accuracy	0.8099002344218503	0.8051572807065366
Sensibilidad	0.7896055103318722	0.7927363807138385
Mayoría de votos		
Accuracy	0.803412746006651	0.7947445892165949
Sensibilidad	0.802128991859737	0.802128991859737

Clasificación

7. SVM

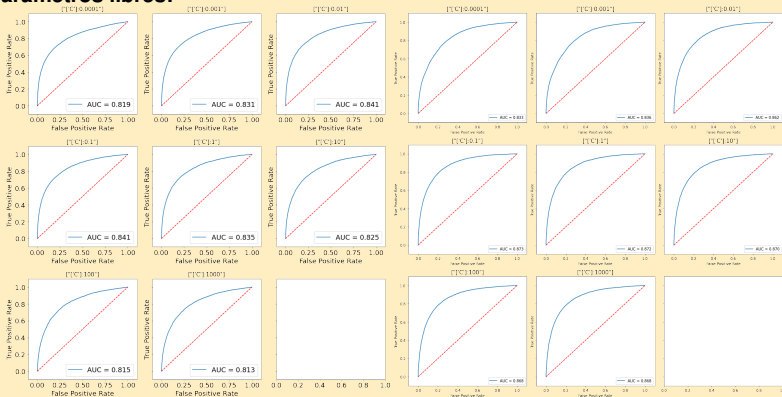
Máquina Vector de soporte: Es un algoritmo que trabaja mal con una cantidad excesiva de datos por lo que se ha cogido un subconjunto randomizado con aproximadamente 1500 datos. Aún habiendo cogido una muestra tan pequeña comparado con los datos de los que disponíamos, los diferentes algoritmos han tardado hasta 7 horas en procesar el conjunto. Se ha hecho:

- SVM Lineal
- SVM Polinómico
- SVM RBF

Clasificación

7. SVM Lineal

Parámetros libres:



Clasificación

7. SVM Lineal desbalanceados

C=1000:

Recall: 0.32591876208897486

[[62773 2926]

[4182 2022]]

C=100:

Recall: 0.33752417794970985

[[62693 3006]

[4110 2094]]

C=10:

Recall: 0.32269503546099293

[[63168 2531]

[4202 2002]]

Clasificación

7. SVM Lineal balanceados

PARÁMETROS LIBRES:

C: 0.0001 sensibility is: 0.0

C: 0.001 sensibility is: 0.5726279925774952

C: 0.01 sensibility is: 0.7138533142700697

C: 0.1 sensibility is: 0.7389651081434612

C: 1 sensibility is: 0.7457183706993581

C: 10 sensibility is: 0.7462811425790162

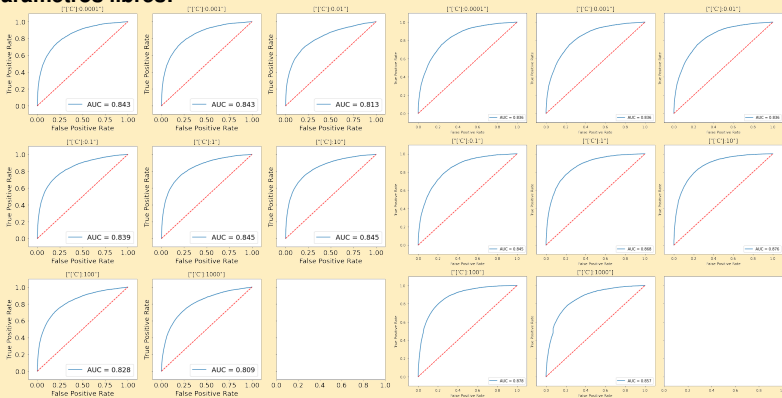
C: 100 sensibility is: 0.7440908952635902

C: 1000 sensibility is: 0.7461442521218021

Clasificación

7. SVM RBF

Parámetros libres:



Clasificación

7. SVM Selección de parámetros libres

Best parameters set found on development set:

```
{'C': 1, 'kernel': 'linear'}
```

Grid scores on development set:

```
0.461 (+/-0.001) for {'C': 1, 'gamma': 0.001, 'kernel': 'rbf'}
0.461 (+/-0.001) for {'C': 1, 'gamma': 0.0001, 'kernel': 'rbf'}
0.461 (+/-0.001) for {'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}
0.461 (+/-0.001) for {'C': 10, 'gamma': 0.0001, 'kernel': 'rbf'}
0.461 (+/-0.001) for {'C': 100, 'gamma': 0.001, 'kernel': 'rbf'}
0.461 (+/-0.001) for {'C': 100, 'gamma': 0.0001, 'kernel': 'rbf'}
0.461 (+/-0.001) for {'C': 1000, 'gamma': 0.001, 'kernel': 'rbf'}
0.461 (+/-0.001) for {'C': 1000, 'gamma': 0.0001, 'kernel': 'rbf'}
0.714 (+/-0.106) for {'C': 1, 'kernel': 'linear'}
0.684 (+/-0.083) for {'C': 10, 'kernel': 'linear'}
0.665 (+/-0.087) for {'C': 100, 'kernel': 'linear'}
0.683 (+/-0.066) for {'C': 1000, 'kernel': 'linear'}
```

Detailed classification report:

The model is trained on the full development set.

The scores are computed on the full evaluation set.

	precision	recall	f1-score	support
0.0	0.94	0.97	0.95	65699
1.0	0.46	0.31	0.37	6204
accuracy			0.91	71903
macro avg	0.70	0.64	0.66	71903
weighted avg	0.90	0.91	0.90	71903

Tuning hyper-parameters for recall

Resultados

Conclusión

Mejores prestaciones de los clasificadores robustos y/o lineales:

- LDA (Oversampling y Unbalanced → Muy rápido)
- Mayoría de votos (Undersampling)
- Boosted Trees (Oversampling)
- Random Forest

Además...

- Destacar la importancia de la calidad de los datos (GiGo), el balanceo para mejorar las prestaciones.
- No hemos obtenido mejoras con la selección de características

Esquema de la Presentación

1 Presentacion

2 **Tutorial**

Tutorial

Tutorial

- El código se encuentra en un repositorio público de GitHub y con acceso a los notebooks en Google Colab.

GitHub Link

- Link directo a GitHub: [ICU Mortality Prediction](#)