

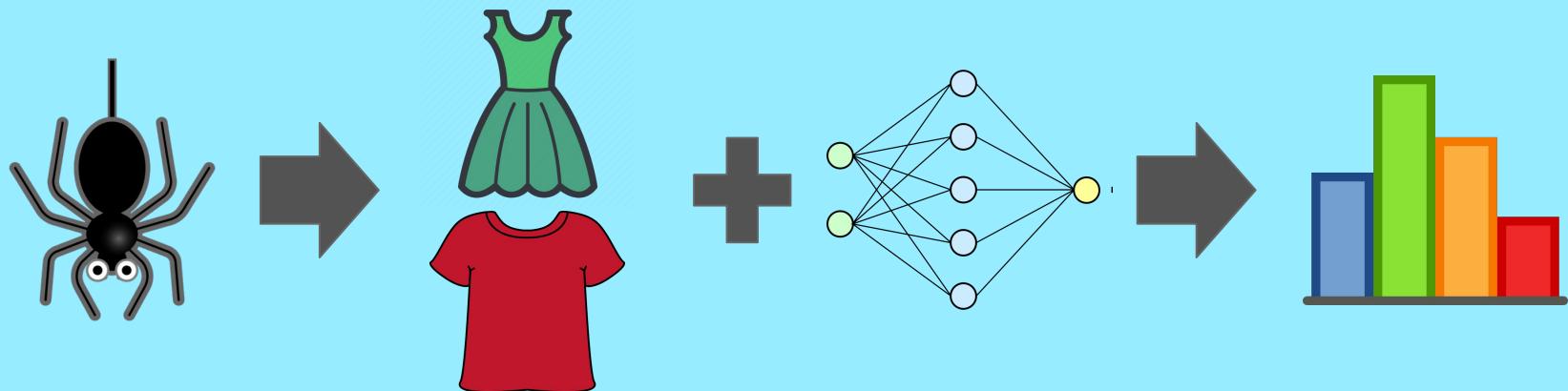
`assertTrue('Testing en Machine Learning')`

Beatriz Gómez

#PyConEs19

Beatriz Gomez
@G_Beaa

STYLE SAGE



¿QUÉ VAMOS A HACER? ¡¡TESTS!!



¿QUÉ VAMOS A HACER? ¡¡TESTS!!

```
@pytest.hookimpl(tryfirst=True, hookwrapper=True)
def pytest_runtest_makereport(item, call):
    # execute all other hooks to obtain the report object
    outcome = yield
    rep = outcome.get_result()

    # set a report attribute for each phase of a call, which can
    # be "setup", "call", "teardown"

    setattr(item, "rep_" + rep.when, rep)

@pytest.fixture
def something(request):
    yield
    # request.node is an "item" because we use the default
    # "function" scope
    if request.node.rep_setup.failed:
        print("setting up a test failed!", request.node.nodeid)
    elif request.node.rep_setup.passed:
        if request.node.rep_call.failed:
            print("executing test failed", request.node.nodeid)

@pytest.mark.xfail(strict=True)
def test_where_failure_is_mandatory():
    # Requires failed assertions! (returns "xfail" on failure; FAILs on pass!)
    assert True

    def test_with_data_fixture(one_fixture):
        """
        PyTest finds the fixture whose name matches the argument,
        calls it, and passes that return value into our test case:
        """
        print("\nRunning test_with_data_fixture: {}".format(one_fixture))
        assert one_fixture == 1
```

```
@pytest.fixture()
def create_book():
    book_data = {
        'title': 'meetup 12 2015',
        'isbn': '1112223334445',
        'value': Decimal('132.23')
    }
    return Book.objects.create(**book_data)

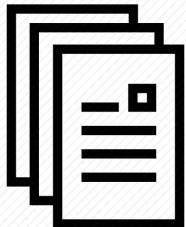
def test_unicode(create_book):
    book = create_book
    assert book.__unicode__() == "The book {} costs: {}".format(
        "meetup 12 2015", Decimal('132.23'))

import pytest
|
@pytest.mark.skip
def test_broken_feature():
    # Always skipped!
    assert False

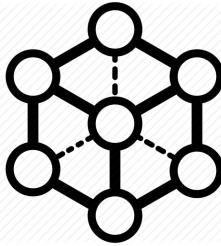
@pytest.mark.skipif(not dev_s3_credentials, reason="S3 creds not found!")
def test_s3_api():
    # Skipped if a certain condition is met
    assert True

@pytest.mark.xfail
def test_where_failure_is_acceptable():
    # Allows failed assertions (returns "XPASS" if there are no failures)
    assert True
```

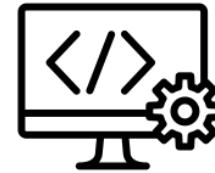
LOS DATA SCIENTIST TAMBIÉN TESTEAMOS!!



DATOS

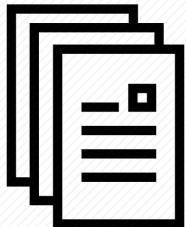


MODELO

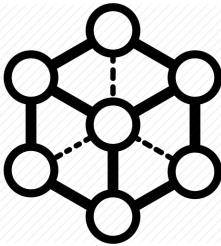


FRAMEWORK

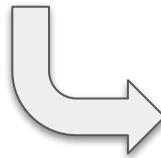
LOS DATA SCIENTIST TAMBIÉN TESTEAMOS!!



DATOS



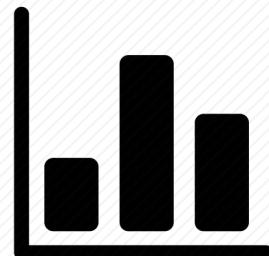
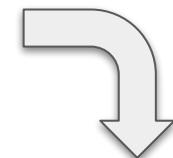
MODELO



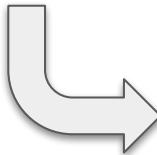
Procesamiento
de los datos

Definición del
modelo

Entrenamiento



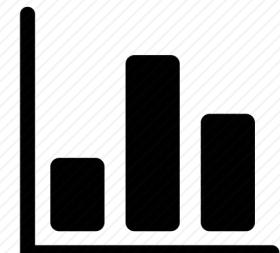
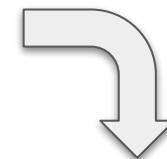
ML PIPELINE



Procesamiento
de los datos

Definición del
modelo

Entrenamiento



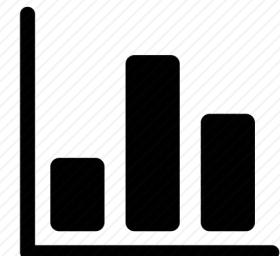
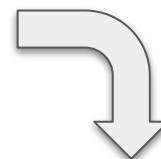
ML PIPELINE



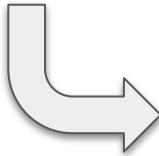
Procesamiento
de los datos

Definición del
modelo

Entrenamiento



ML PIPELINE



Procesamiento
de los datos

Definición del
modelo

Entrenamiento



ML PIPELINE



Procesamiento datos



Definir modelo



Entrenamiento



Preprocesamiento de los datos





Procesamiento datos



Definir modelo



Entrenamiento



Checklist procesamiento básico

- Detectar NaN/nulos.
- Detectar cadenas vacías.
- Valores fuera de rango. #sensores
- Normalización. #sensores
- Uppercase/lowercase. #texto
- Imágenes vacías /blancas /negras /pequeñas. #imágenes



Procesamiento datos



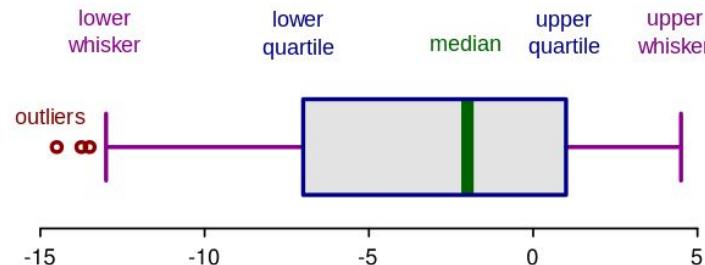
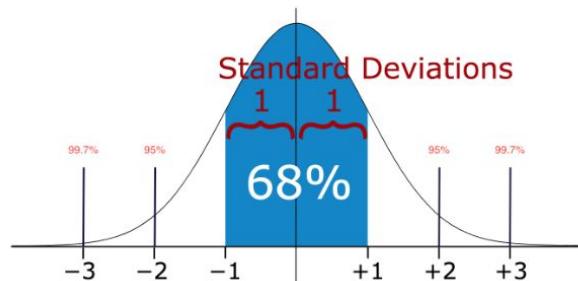
Definir modelo



Entrenamiento



Detección de outliers



Desviaciones estadísticas

Boxplots



Procesamiento datos



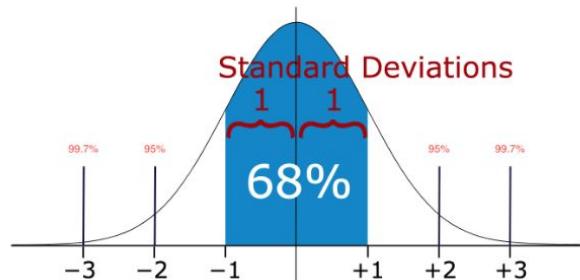
Definir modelo



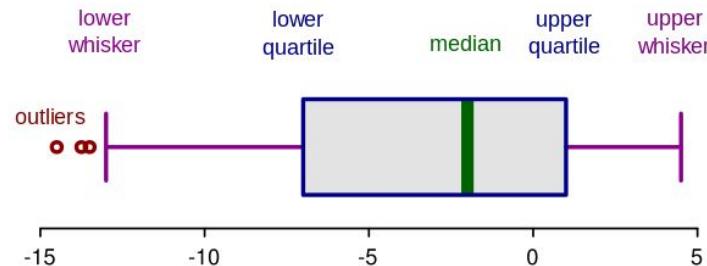
Entrenamiento



Detección de outliers



Desviaciones estadísticas



Boxplots



Procesamiento datos



Definir modelo



Entrenamiento



Detección de outliers



Correlación entre feature-labels



Procesamiento datos



Definir modelo



Entrenamiento

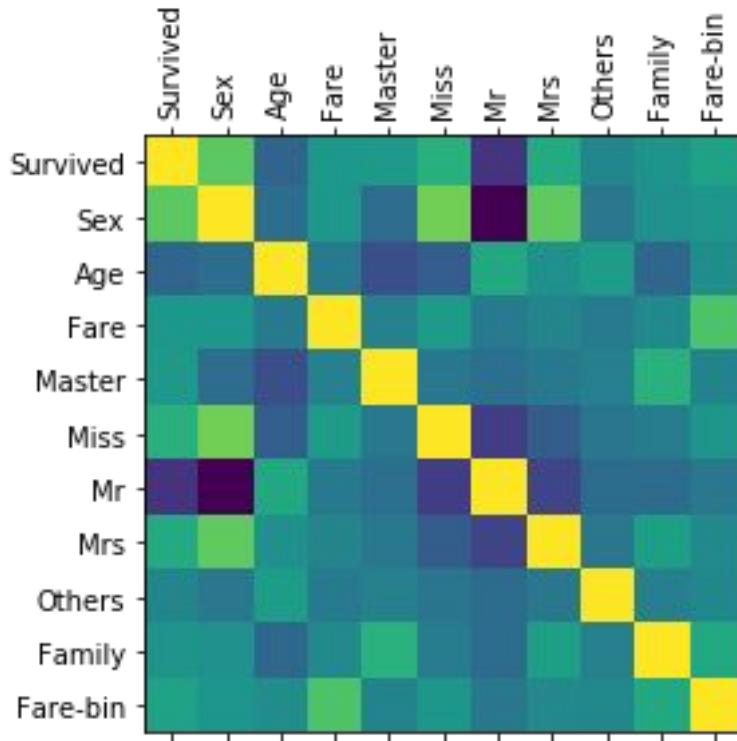


Detección de outliers



Correlación entre feature-labels

```
import matplotlib.pyplot as plt  
  
plt.matshow(df_train.corr())  
tick_marks = [i for i in range(len(df_train.columns))]  
plt.xticks(tick_marks, df_train.columns, rotation='vertical')  
plt.yticks(tick_marks, df_train.columns)  
plt.show()
```





Procesamiento datos



Definir modelo



Entrenamiento

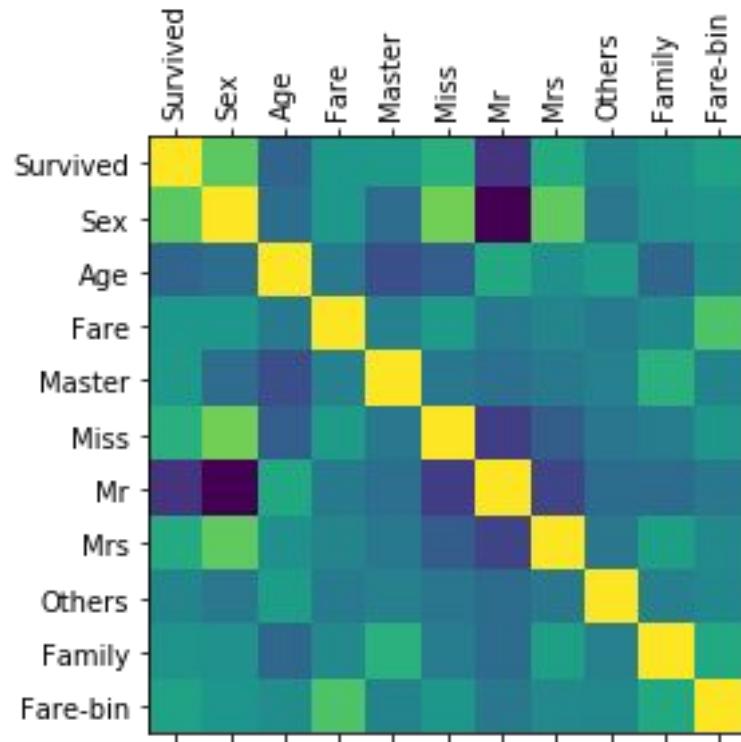


Detección de outliers



Correlación entre feature-labels

```
import matplotlib.pyplot as plt  
  
plt.matshow(df_train.corr())  
tick_marks = [i for i in range(len(df_train.columns))]  
plt.xticks(tick_marks, df_train.columns, rotation='vertical')  
plt.yticks(tick_marks, df_train.columns)  
plt.show()
```





Procesamiento datos



Definir modelo



Entrenamiento



Detección de outliers



Correlación entre feature-labels



Detección de valores duplicados



Procesamiento datos



Definir modelo



Entrenamiento



Detección de outliers



Correlación entre feature-labels



Detección de valores duplicados

```
1 | DataFrame.duplicated(subset=None, keep='first')
```

```
>>> hashlib.sha224(b"Nobody inspects the spammish repetition").hexdigest()
'a4337bc45a8fc544c03f52dc550cd6e1e87021bc896588bd79e901e2'
```



Procesamiento datos



Definir modelo



Entrenamiento



Detección de outliers



Correlación entre feature-labels



Detección de valores duplicados

```
1 | DataFrame.duplicated(subset=None, keep='first')
```

```
>>> hashlib.sha224(b"Nobody inspects the spammish repetition").hexdigest()
'a4337bc45a8fc544c03f52dc550cd6e1e87021bc896588bd79e901e2'
```



Procesamiento datos



Definir modelo



Entrenamiento



Detección de outliers



Correlación entre feature-labels



Detección de valores duplicados



Balanceo de clases



Procesamiento datos



Definir modelo



Entrenamiento



- Detección de outliers
- Correlación entre feature-labels
- Detección de valores duplicados
- Balanceo de clases





Procesamiento datos



Definir modelo



Entrenamiento



- Detección de outliers
- Correlación entre feature-labels
- Detección de valores duplicados
- Balanceo de clases





Procesamiento datos



Definir modelo



Entrenamiento



Detección de outliers



Correlación entre feature-labels



Detección de valores duplicados



Balanceo de clases



Distribución de los splits



Procesamiento datos



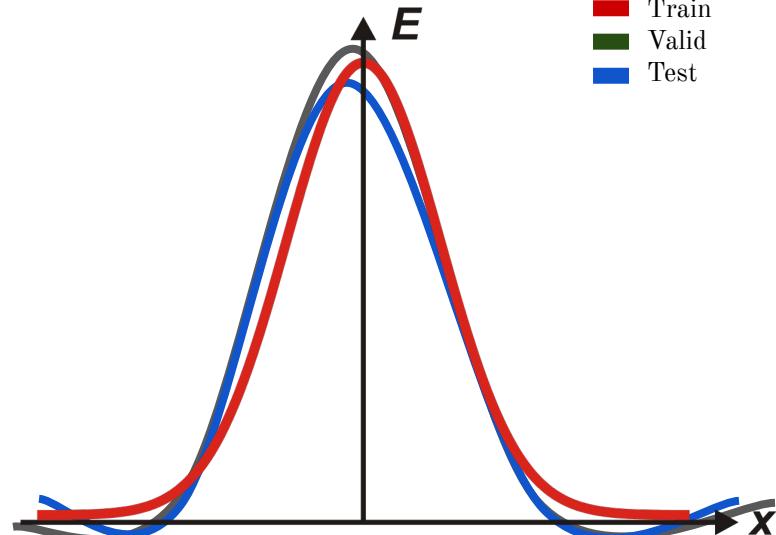
Definir modelo



Entrenamiento



- Detección de outliers
- Correlación entre feature-labels
- Detección de valores duplicados
- Balanceo de clases
- Distribución de los splits





Procesamiento datos



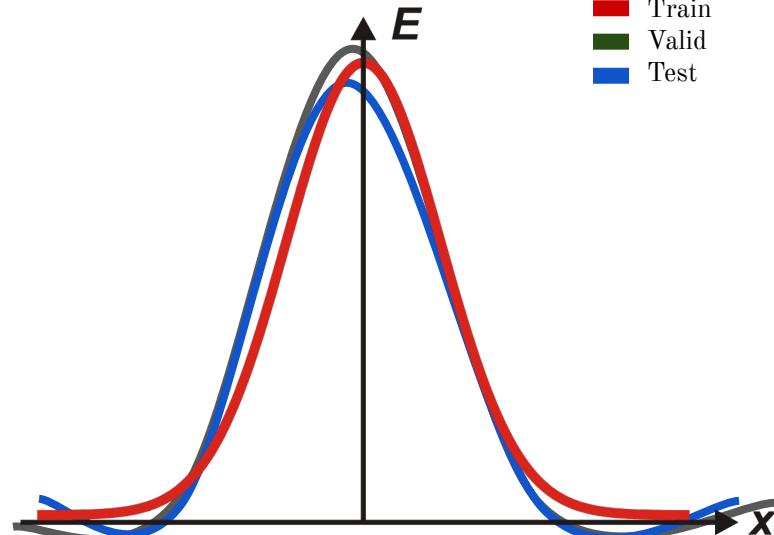
Definir modelo



Entrenamiento



- Detección de outliers
- Correlación entre feature-labels
- Detección de valores duplicados
- Balanceo de clases
- Distribución de los splits





Procesamiento datos



Definir modelo



Entrenamiento



Detección de outliers



Correlación entre feature-labels



Detección de valores duplicados



Balanceo de clases



Distribución de los splits



Distribución en las series temporales
(ventana deslizante)



Procesamiento datos



Definir modelo



Entrenamiento



Detección de outliers



Correlación entre feature-labels



Detección de valores duplicados



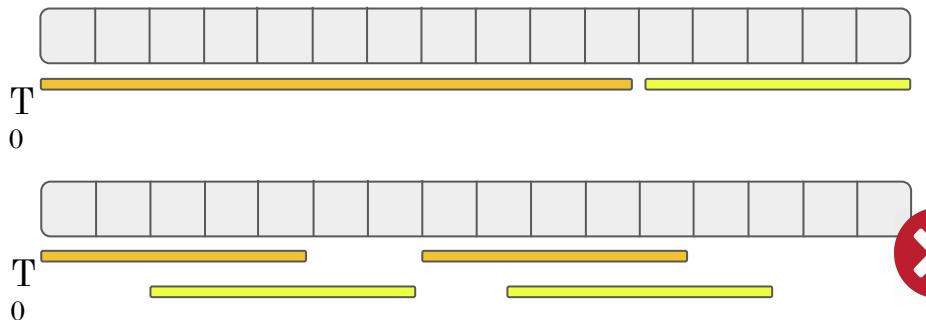
Balanceo de clases



Distribución de los splits



Distribución en las series temporales
(ventana deslizante)



Train
Test



Procesamiento datos



Definir modelo



Entrenamiento



Detección de outliers



Correlación entre feature-labels



Detección de valores duplicados



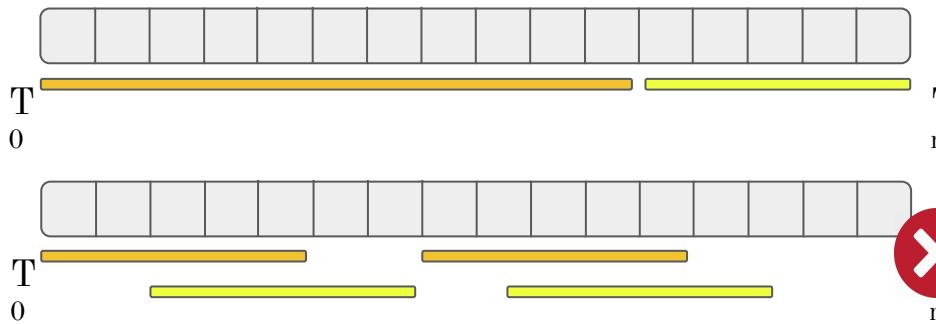
Balanceo de clases



Distribución de los splits



Distribución en las series temporales
(ventana deslizante)



Train
Test



Procesamiento datos



Definir modelo



Entrenamiento



Definimos el modelo





Procesamiento datos



Definir modelo



Entrenamiento



Checklist definición del modelo



Sobreentrenar el modelo



Procesamiento datos



Definir modelo



Entrenamiento

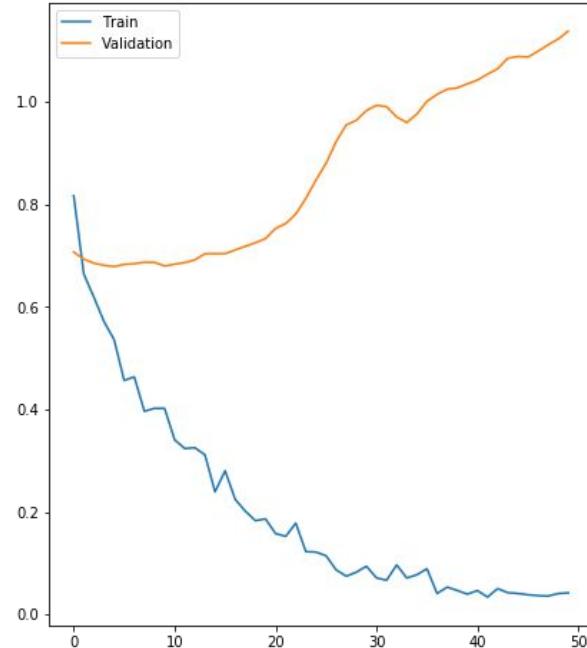


Checklist definición del modelo



Sobreentrenar el modelo

```
training = model.fit(x_train[:40], y_train[:40],  
                      epochs=100,  
                      batch_size=16,  
                      validation_split=0.2,  
                      verbose=2)
```





Procesamiento datos



Definir modelo



Entrenamiento

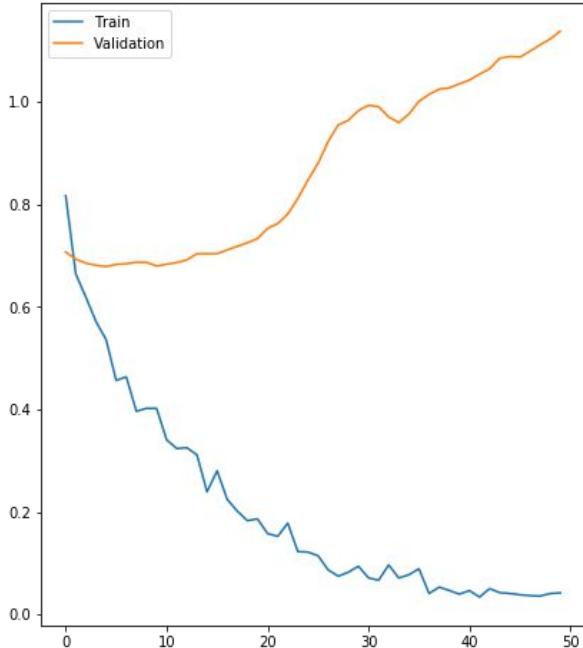


Checklist definición del modelo



Sobreentrenar el modelo

```
training = model.fit(x_train[:40], y_train[:40],  
                      epochs=100,  
                      batch_size=16,  
                      validation_split=0.2,  
                      verbose=2)
```





Procesamiento datos



Definir modelo



Entrenamiento



Checklist definición del modelo

- Sobreentrenar el modelo
- Introducimos información “artificial”



Procesamiento datos



Definir modelo

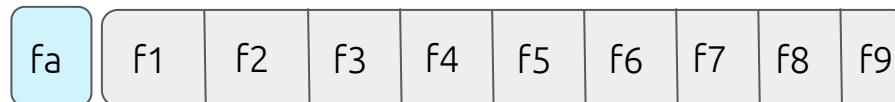


Entrenamiento



Checklist definición del modelo

- Sobreentrenar el modelo
- Introducimos información “artificial”





Procesamiento datos



Definir modelo



Entrenamiento



Checklist definición del modelo

- Sobreentrenar el modelo
- Introducimos información “artificial”





Procesamiento datos



Definir modelo



Entrenamiento



Checklist definición del modelo

- Sobreentrenar el modelo
- Introducimos información “artificial”
- Hipótesis sobre el *loss* en las NN



Procesamiento datos



Definir modelo



Entrenamiento



Checklist definición del modelo

- Sobreentrenar el modelo
- Introducimos información "artificial"
- Hipótesis sobre el *loss* en las NN

$$\text{Softmax loss} = -\text{Pocurriencia} \cdot \ln(\text{Pclase})$$

- 10 clases balanceadas
- 0.1 por clase
- $\text{Softmax loss} = 10 \cdot (0.1 \cdot \ln(0.1)) = 2.302$



Procesamiento datos



Definir modelo



Entrenamiento



Checklist definición del modelo

- Sobreentrenar el modelo
- Introducimos información "artificial"
- Hipótesis sobre el *loss* en las NN

$$\text{Softmax loss} = -\text{Pocurriencia} \cdot \ln(\text{Pclase})$$

- 10 clases balanceadas
- 0.1 por clase
- $\text{Softmax loss} = 10 \cdot (0.1 \cdot \ln(0.1)) = 2.302$



Procesamiento datos



Definir modelo



Entrenamiento



Checklist definición del modelo

- Sobreentrenar el modelo
- Introducimos información “artificial”
- Hipótesis sobre el *loss* en las NN
- Los minibatches también cuentan



Procesamiento datos



Definir modelo

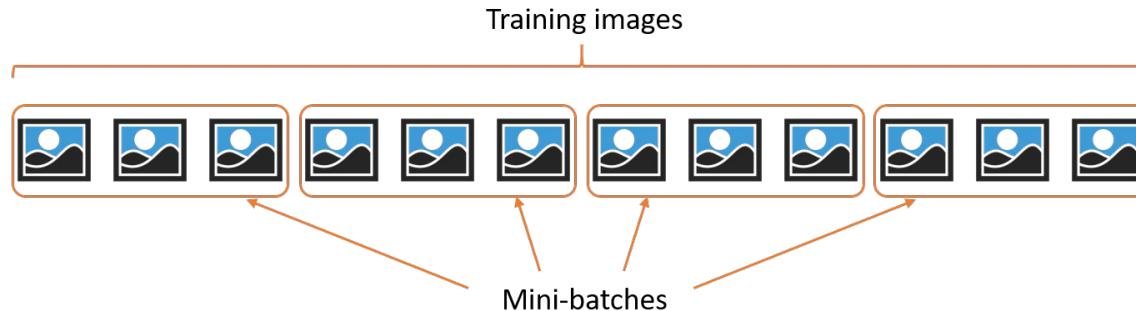


Entrenamiento



Checklist definición del modelo

- Sobreentrenar el modelo
- Introducimos información “artificial”
- Hipótesis sobre el *loss* en las NN
- Los minibatches también cuentan





Procesamiento datos



Definir modelo

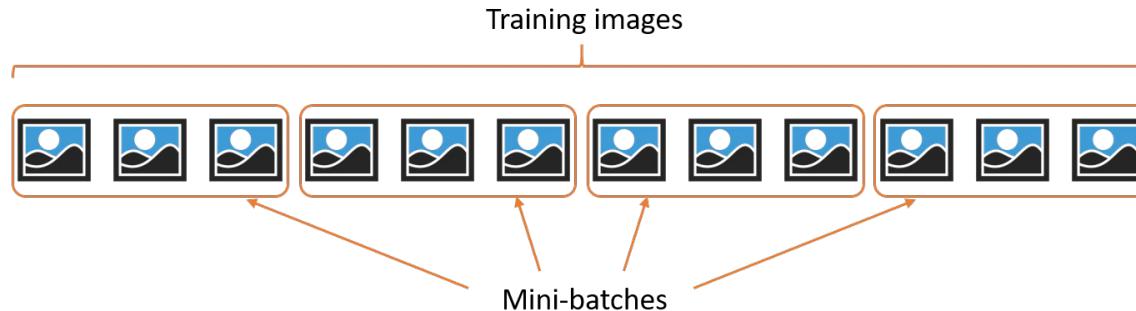


Entrenamiento



Checklist definición del modelo

- Sobreentrenar el modelo
- Introducimos información "artificial"
- Hipótesis sobre el *loss* en las NN
- Los minibatches también cuentan





Procesamiento datos



Definir modelo



Entrenamiento



Aseguramos
el
modelo





Procesamiento datos



Definir modelo



Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches



Procesamiento datos



Definir modelo



Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches

```
random.shuffle(x, random)
```





Procesamiento datos



Definir modelo



Entrenamiento



Checklist comprobación del modelo



Shuffle de los datos y de minibatches

```
random.shuffle(x, random)
```





Procesamiento datos



Definir modelo



Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches
- Shuffle en texto



Procesamiento datos



Definir modelo



Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches
- Shuffle en texto

'Hola'	'soy'	'Beatriz'
--------	-------	-----------



'Hola'	'Beatriz'	'soy'
--------	-----------	-------



Procesamiento datos



Definir modelo



Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches
- Shuffle en texto

'Hola'	'soy'	'Beatriz'
--------	-------	-----------



'Hola'	'Beatriz'	'soy'
--------	-----------	-------



Procesamiento datos



Definir modelo



Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches
- Shuffle en texto
- Modificamos (%) las etiquetas



Procesamiento datos



Definir modelo

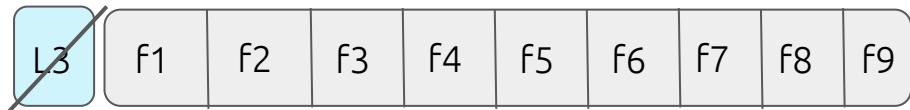


Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches
- Shuffle en texto
- Modificamos (%) las etiquetas





Procesamiento datos



Definir modelo

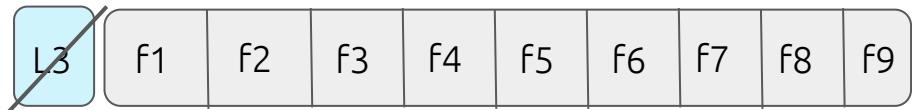


Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches
- Shuffle en texto
- Modificamos (%) las etiquetas





Procesamiento datos



Definir modelo



Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches
- Shuffle en texto
- Modificamos (%) las etiquetas
- Cambiamos palabras en texto



Procesamiento datos



Definir modelo



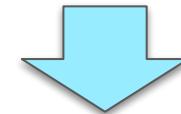
Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches
- Shuffle en texto
- Modificamos (%) las etiquetas
- Cambiamos palabras en texto

'Hola'	'soy'	'Beatriz'
--------	-------	-----------



'Sombrilla'	'soy'	'Beatriz'
-------------	-------	-----------



Procesamiento datos



Definir modelo



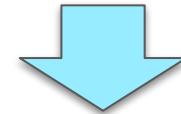
Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches
- Shuffle en texto
- Modificamos (%) las etiquetas
- Cambiamos palabras en texto

'Hola'	'soy'	'Beatriz'
--------	-------	-----------



'Sombrilla'	'soy'	'Beatriz'
-------------	-------	-----------



Procesamiento datos



Definir modelo



Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches
- Shuffle en texto
- Modificamos (%) las etiquetas
- Cambiamos palabras en texto
- Modificar feature mas correlada ¡en caja blanca!



Procesamiento datos



Definir modelo



Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches
- Shuffle en texto
- Modificamos (%) las etiquetas
- Cambiamos palabras en texto
- Modificar feature mas correlada ¡en caja blanca!

```
if age < 10:  
    survived = True
```



Procesamiento datos



Definir modelo



Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches
- Shuffle en texto
- Modificamos (%) las etiquetas
- Cambiamos palabras en texto
- Modificar feature mas correlada ¡en caja blanca!

```
if age < 10:  
    survived = True
```



Procesamiento datos



Definir modelo



Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches
- Shuffle en texto
- Modificamos (%) las etiquetas
- Cambiamos palabras en texto
- Modificar feature mas correlada ¡en caja blanca!
- Ablation study* en NN



Procesamiento datos



Definir modelo

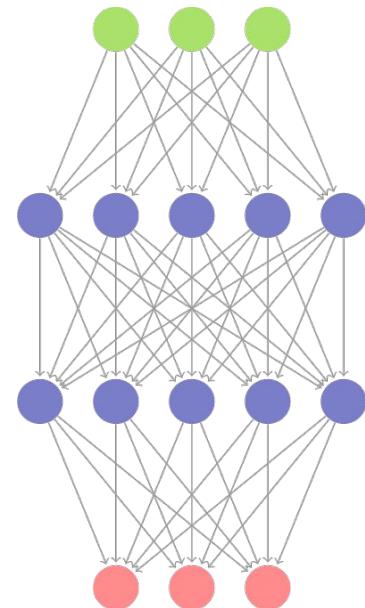
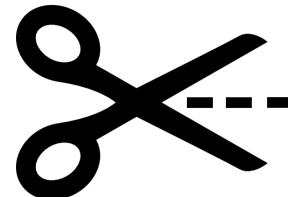


Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches
- Shuffle en texto
- Modificamos (%) las etiquetas
- Cambiamos palabras en texto
- Modificar feature mas correlada ¡en caja blanca!
- Ablation study* en NN





Procesamiento datos



Definir modelo

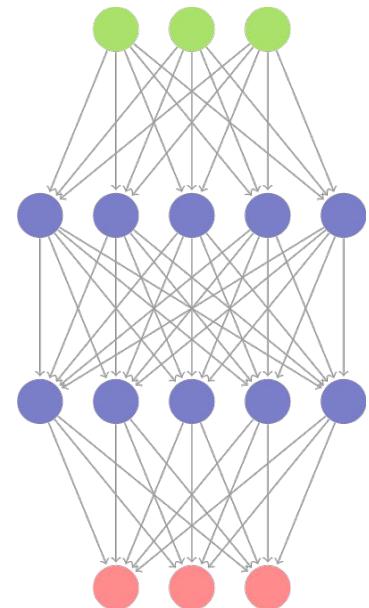
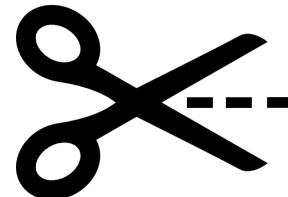


Entrenamiento



Checklist comprobación del modelo

- Shuffle de los datos y de minibatches
- Shuffle en texto
- Modificamos (%) las etiquetas
- Cambiamos palabras en texto
- Modificar feature mas correlada ¡en caja blanca!
- Ablation study* en NN



Conclusiones

1. Los test de SW no cubren todos los errores en ML
2. Elegir los test que nos ayuden
3. Hay que compartir

#MiTestEnML

GRACIAS



@g_beaa



beatriz@stylesage.co