Detección de anomalías

Versión 3

Librerias

In [12]:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.graphics.tsaplots import plot_pacf
from math import nan, isnan
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures

# Import rpy2 for dataframe conversion
import rpy2.robjects as ro
from rpy2.robjects.packages import importr
from rpy2.robjects import pandas2ri
from rpy2.robjects.conversion import localconverter
from rpy2.robjects import globalenv
```

Cargando los datos

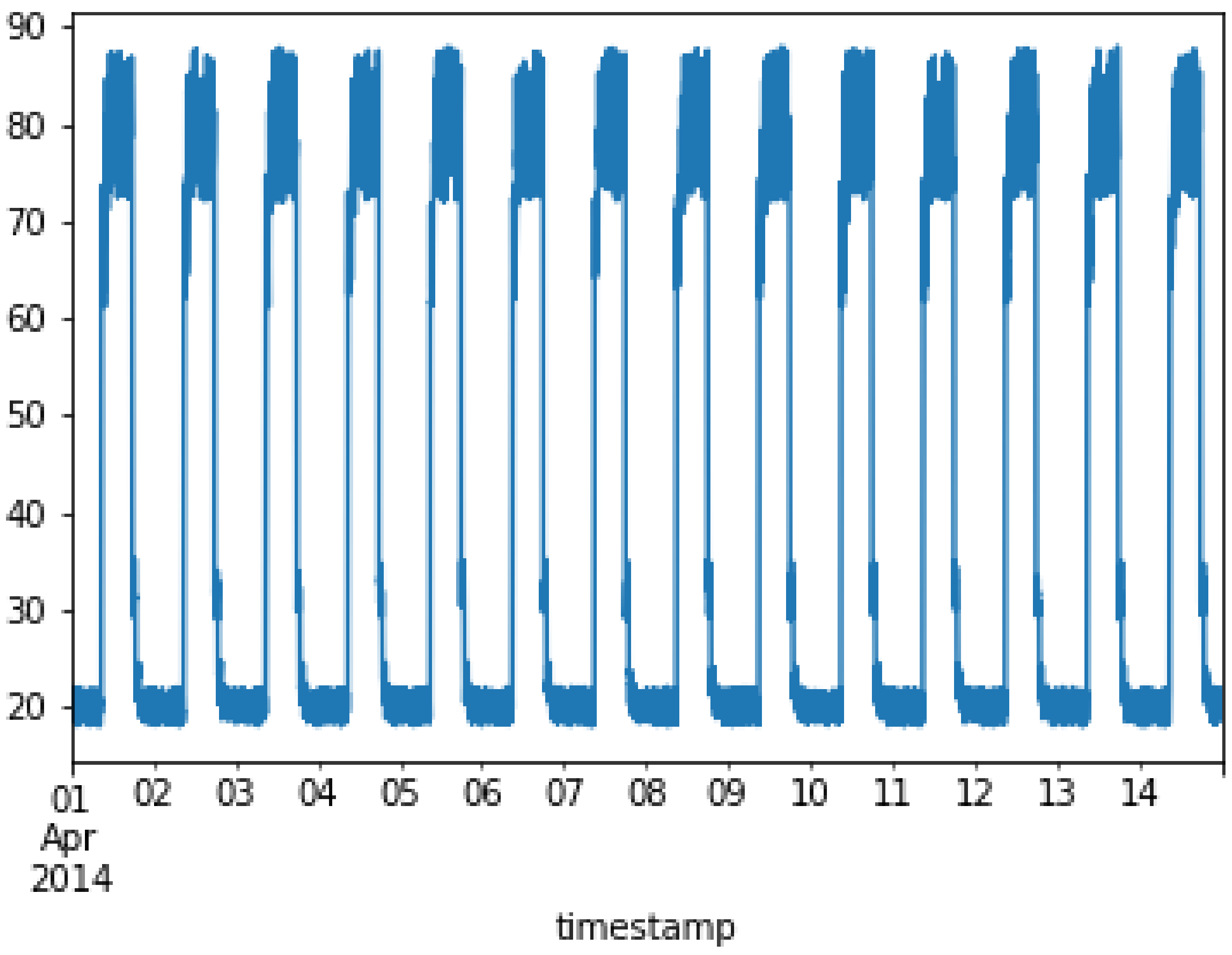
In [13]:

```
master_url_root = "https://raw.githubusercontent.com/numenta/NAB/master/data/"

df_small_noise_url_suffix = "artificialNoAnomaly/art_daily_small_noise.csv"
df_small_noise_url = master_url_root + df_small_noise_url_suffix
df_small_noise = pd.read_csv(
    df_small_noise_url, parse_dates=True, index_col="timestamp"
)

df_daily_jumpsup_url_suffix = "artificialWithAnomaly/art_daily_jumpsup.csv"
df_daily_jumpsup_url = master_url_root + df_daily_jumpsup_url_suffix
df_daily_jumpsup = pd.read_csv(
    df_daily_jumpsup_url, parse_dates=True, index_col="timestamp"
)
```

Patron regular

In [14]:

```
fig, ax = plt.subplots()
df_small_noise.plot(legend=False, ax=ax)
plt.show()
df_small_noise.size
```



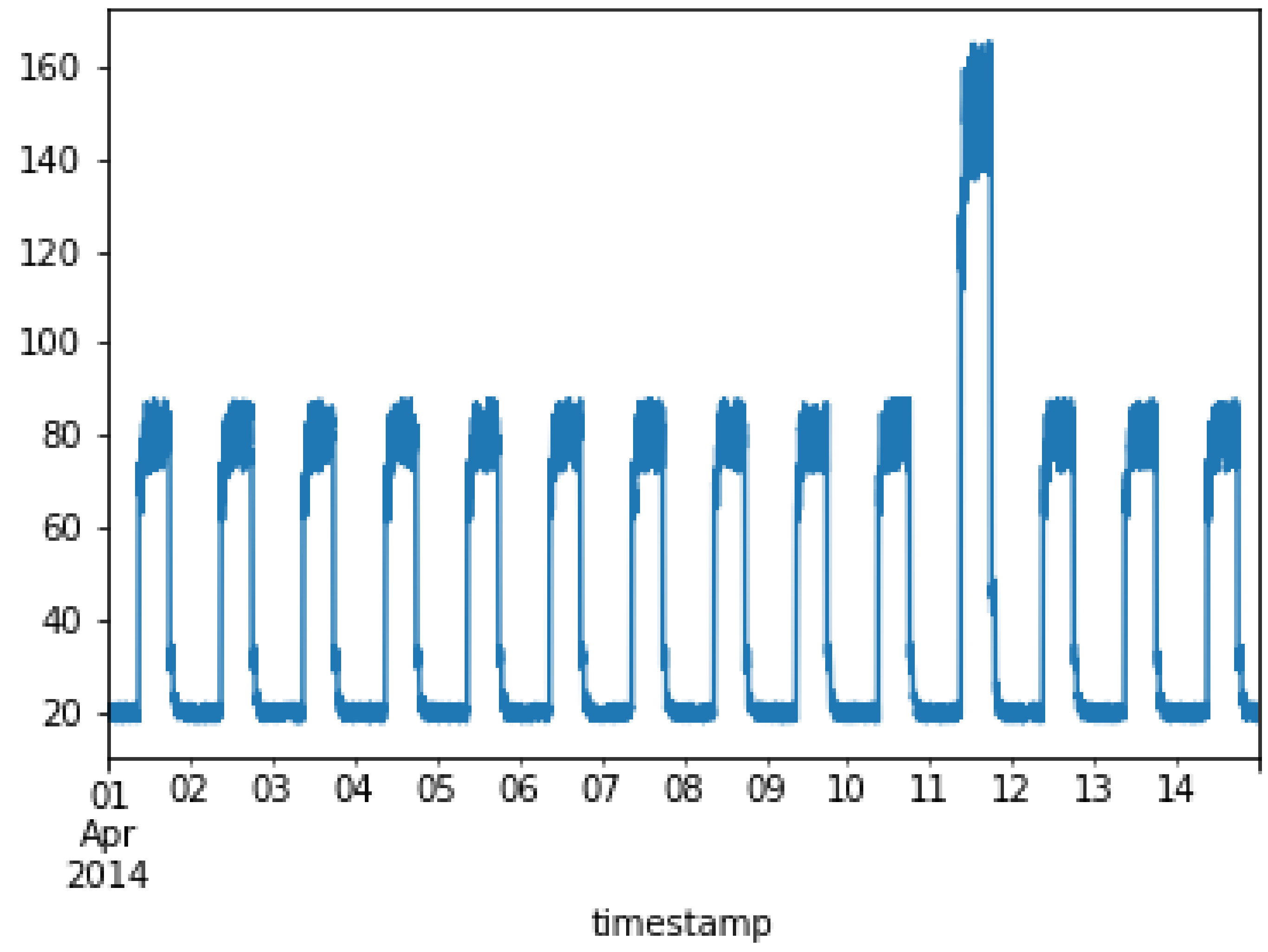
Out[14]:

```
4032
```

Patron anormal

In [15]:

```
fig, ax = plt.subplots()
df_daily_jumpsup.plot(legend=False, ax=ax)
plt.show()
```



Extrayendo una muestra de 2000 observaciones.

In [16]:

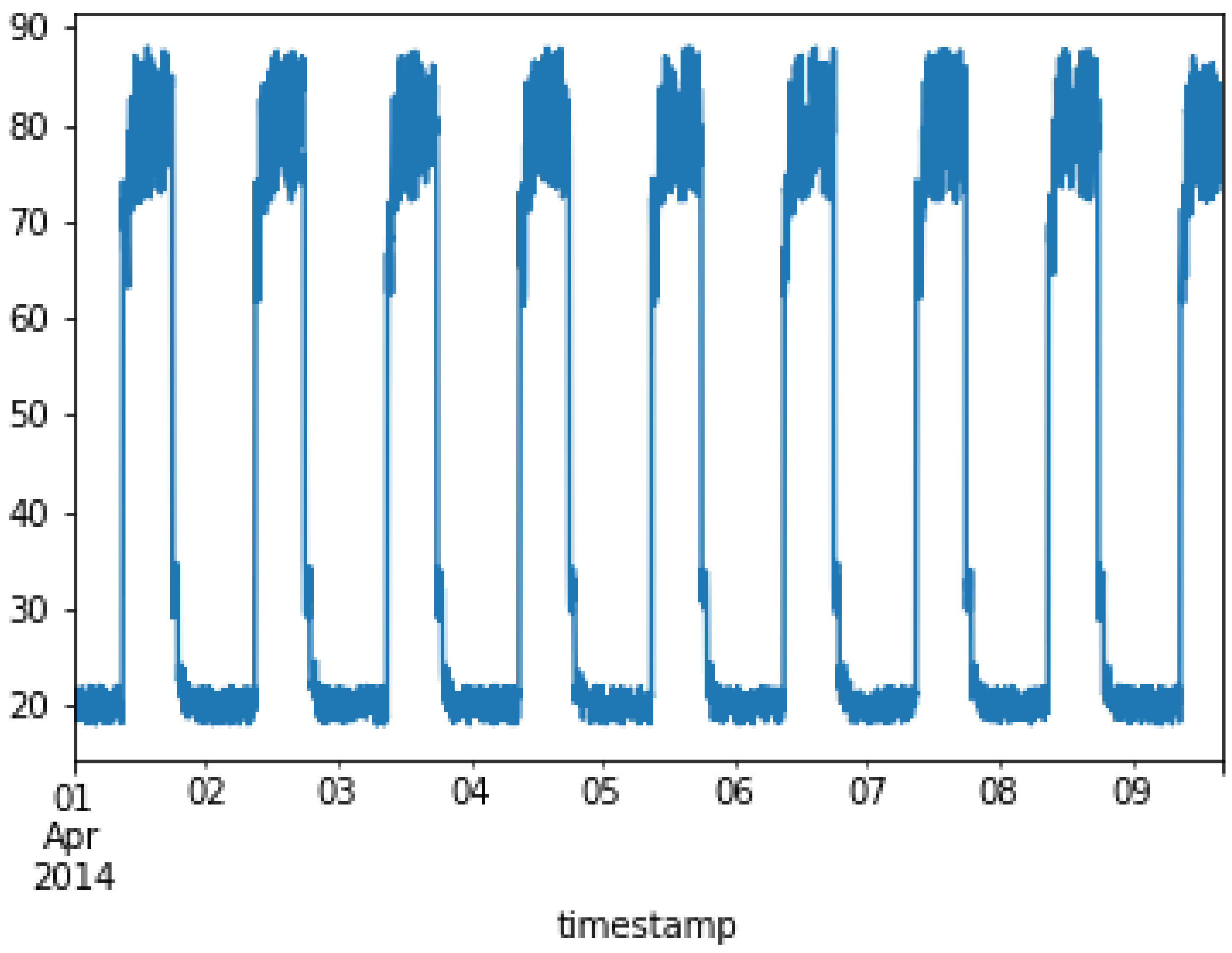
```
jump = df_daily_jumpsup.iloc[0:2500]
jump.size
```

Out[16]:

```
2500
```

In [17]:

```
fig, ax = plt.subplots()
jump.plot(legend=False, ax=ax)
plt.show()
```



Análisis

Un estimador de la varianza.

1. Se construirá un estimador de la varianza a partir de una regresión polinomial local de grado P en una vecindad de radio R.
2. El estimador de la varianza es la varianza estimada del polinomio.

Selección

In [18]:

```
ST = jump
```

Análizando al media y volatilidad de los datos

In [19]:

```
ST.to_csv("/content/datos.csv")
```

In [27]:

```
colnames=['timestamp','Y','media','radio','Y1','Y109','Y181','Y289']
datos = pd.read_csv('/content/datos2.csv',
                    names=colnames,
                    header=1,
                    sep=';')
datos
```

Out[27]:

	timestamp	Y	media	radio	Y1	Y109	Y1
2	2014-04-01 00:05:00	20,500833287	20,094374190288	1,10508924688327	19,761251903	NaN	Na
3	2014-04-01 00:10:00	19,9616414445	20,094374190288	1,10508924688327	20,500833287	NaN	Na
4	2014-04-01 00:15:00	21,4902660734	20,094374190288	1,10508924688327	19,9616414445	NaN	Na
5	2014-04-01 00:20:00	20,1877394098	20,094374190288	1,10508924688327	21,4902660734	NaN	Na
6	2014-04-01 00:25:00	19,9231256718	20,094374190288	1,10508924688327	20,1877394098	NaN	Na
...	...	...	...	...	...	...	
2496	2014-04-09 15:55:00	83,6180952106	79,2892291169724	4,66709829633724	79,1160883348	20,3112860251	20,47214992
2497	2014-04-09 16:00:00	82,9164347484	79,2892291169724	4,66709829633724	83,6180952106	18,4550468868	20,82225276
2498	2014-04-09 16:05:00	74,1656569898	79,2892291169724	4,66709829633724	82,9164347484	19,9078715644	21,31528902
2499	2014-04-09 16:10:00	74,2860643176	79,2892291169724	4,66709829633724	74,1656569898	19,42490755	21,51067707
2500	2014-04-09 16:15:00	84,0853108028	79,2892291169724	4,66709829633724	74,2860643176	19,9641538794	20,57055868

2499 rows × 8 columns

Quitando las filas con valores faltantes.

In [38]:

```
datos2 = datos.dropna()
datos2.head()
```

Out[38]:

	timestamp	Y	media	radio	Y1	Y109	Y18
290	2014-04-02 00:05:00	20,1630729313	20,3325890728322	1,98982373263666	21,6086167546	77,0638638812	74,126014383
291	2014-04-02 00:10:00	20,7998376509	20,3325890728322	1,98982373263666	20,1630729313	84,51222102	69,103611643

	timestamp	Y	media	radio	Y1	Y109	Y18
292	2014-04-02 00:15:00	21,3124340371	20,3325890728322	1,98982373263666	20,7998376509	74,8044287246	72,280854066
293	2014-04-02 00:20:00	21,1661640253	20,3325890728322	1,98982373263666	21,3124340371	84,5211212147	66,954407486
294	2014-04-02 00:25:00	21,5304224949	20,3325890728322	1,98982373263666	21,1661640253	82,6201685228	73,485079478

Se hace una partición de 70% para entrenamiento y 30% para validación.

In [32]:

```
from sklearn.model_selection import train_test_split

Xtrain, Xtest, Ytrain, Ytest = train_test_split(datos2[['Y1','Y109','Y181','Y289']],
                                                datos2[['media','radio']],
                                                test_size = 0.30,
                                                random_state = 0)
```

In [37]:

```
Xtrain.head()
```

Out[37]:

	Y1	Y109	Y181	Y289
1284	81,0744785146	21,8764661238	21,812735196	80,4205419761
977	67,3754355626	21,8627556662	29,8841486663	62,5319617692
1647	83,4222314751	18,123011762	20,1580917741	80,2269800876
2071	19,2803724363	22,1144755508	84,7468042187	19,4875919623
588	18,9253132492	87,4730991023	65,7082949163	18,2826727612

In [39]:

```
Ytrain.head()
```

Out[39]:

	media	radio
1284	79,3676626221598	4,65568428415253
977	68,2947234165003	4,95902421549402
1647	79,7077768021834	4,58707644473642
2071	20,2961389673661	1,99991415644588
588	20,3593718607417	2,13215395921344

Especificación del modelo

In [33]:

```
import tensorflow as tf
```

In [34]:

```
entrada =  tf.keras.layers.Dense(units = 4, activation = "relu", input_shape = [4])
oculta1 = tf.keras.layers.Dense(units = 5, activation = "relu")
oculta2 = tf.keras.layers.Dense(units = 3, activation = "relu")
salida = tf.keras.layers.Dense(units = 2, activation = "relu")
modelo = tf.keras.Sequential([entrada, oculta1, oculta2, salida])
```

In [40]:

```
modelo.compile(
    optimizer = tf.keras.optimizers.Adam(0.1),
    loss = 'mean_squared_error'
)
```

Entrenamiento

In [49]:

```
type(Xtrain)
```

Out[49]:

```
pandas.core.frame.DataFrame
```

In [54]:

```
type(Xtrain.values)
```

Out[54]:

```
numpy.ndarray
```

In [45]:

```
Xtrain.shape
```

Out[45]:

```
(1547, 4)
```

In [56]:

```
Xtrain.dtypes
```

Out[56]:

```
Y1      object
Y109    object
Y181    object
Y289    object
dtype: object
```

In [58]:

```
Xtrain.to_numeric()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-58-30ac763b6b64> in <module>
----> 1 Xtrain.to_numeric()

/usr/local/lib/python3.7/dist-packages/pandas/core/generic.py in __getattr__(self, name)
   5485         ):
   5486             return self[name]
-> 5487         return object.__getattribute__(self, name)
   5488 
   5489     def __setattr__(self, name: str, value) -> None:

AttributeError: 'DataFrame' object has no attribute 'to_numeric'
```

In [55]:

```
historial = modelo.fit(Xtrain.values, Ytrain.values, epochs = 1000, verbose = False)
```

```
---------------------------------------------------------------------------
UnimplementedError                        Traceback (most recent call last)
<ipython-input-55-5d66cecbfe9d> in <module>
----> 1 historial = modelo.fit(Xtrain.values, Ytrain.values, epochs = 1000, verbose = Fals
e)

/usr/local/lib/python3.7/dist-packages/keras/utils/traceback_utils.py in error_handler(*ar
gs, **kwargs)
     65     except Exception as e:  # pylint: disable=broad-except
     66       filtered_tb = _process_traceback_frames(e.__traceback__)
---> 67       raise e.with_traceback(filtered_tb) from None
     68     finally:
     69       del filtered_tb

/usr/local/lib/python3.7/dist-packages/tensorflow/python/eager/execute.py in quick_execute
(op_name, num_outputs, inputs, attrs, ctx, name)
     53     ctx.ensure_initialized()
     54     tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name, op_name,
---> 55                                         inputs, attrs, num_outputs)
     56   except core._NotOkStatusException as e:
     57     if name is not None:

UnimplementedError: Graph execution error:
```

```
Detected at node 'sequential/Cast' defined at (most recent call last):
    File "/usr/lib/python3.7/runpy.py", line 193, in _run_module_as_main
      "__main__", mod_spec)
    File "/usr/lib/python3.7/runpy.py", line 85, in _run_code
      exec(code, run_globals)
    File "/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py", line 16, in <modu
le>
      app.launch_new_instance()
    File "/usr/local/lib/python3.7/dist-packages/traitlets/config/application.py", line 84
6, in launch_instance
      app.start()
    File "/usr/local/lib/python3.7/dist-packages/ipykernel/kernelapp.py", line 612, in sta
rt
      self.io_loop.start()
    File "/usr/local/lib/python3.7/dist-packages/tornado/platform/asyncio.py", line 149, i
n start
      self.asyncio_loop.run_forever()
    File "/usr/lib/python3.7/asyncio/base_events.py", line 541, in run_forever
      self._run_once()
    File "/usr/lib/python3.7/asyncio/base_events.py", line 1786, in _run_once
      handle._run()
    File "/usr/lib/python3.7/asyncio/events.py", line 88, in _run
      self._context.run(self._callback, *self._args)
    File "/usr/local/lib/python3.7/dist-packages/tornado/ioloop.py", line 690, in <lambda>
      lambda f: self._run_callback(functools.partial(callback, future))
    File "/usr/local/lib/python3.7/dist-packages/tornado/ioloop.py", line 743, in _run_cal
lback
      ret = callback()
    File "/usr/local/lib/python3.7/dist-packages/tornado/gen.py", line 787, in inner
      self.run()
    File "/usr/local/lib/python3.7/dist-packages/tornado/gen.py", line 748, in run
      yielded = self.gen.send(value)
    File "/usr/local/lib/python3.7/dist-packages/ipykernel/kernelbase.py", line 365, in pr
ocess_one
      yield gen.maybe_future(dispatch(*args))
    File "/usr/local/lib/python3.7/dist-packages/tornado/gen.py", line 209, in wrapper
      yielded = next(result)
    File "/usr/local/lib/python3.7/dist-packages/ipykernel/kernelbase.py", line 268, in di
spatch_shell
      yield gen.maybe_future(handler(stream, idents, msg))
    File "/usr/local/lib/python3.7/dist-packages/tornado/gen.py", line 209, in wrapper
      yielded = next(result)
    File "/usr/local/lib/python3.7/dist-packages/ipykernel/kernelbase.py", line 545, in ex
ecute_request
      user_expressions, allow_stdin,
    File "/usr/local/lib/python3.7/dist-packages/tornado/gen.py", line 209, in wrapper
      yielded = next(result)
    File "/usr/local/lib/python3.7/dist-packages/ipykernel/ipkernel.py", line 306, in do_e
xecute
      res = shell.run_cell(code, store_history=store_history, silent=silent)
    File "/usr/local/lib/python3.7/dist-packages/ipykernel/zmqshell.py", line 536, in run_
cell
      return super(ZMQInteractiveShell, self).run_cell(*args, **kwargs)
    File "/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py", line 2
855, in run_cell
      raw_cell, store_history, silent, shell_futures)
    File "/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py", line 2
881, in _run_cell
      return runner(coro)
    File "/usr/local/lib/python3.7/dist-packages/IPython/core/async_helpers.py", line 68,
 in _pseudo_sync_runner
      coro.send(None)
    File "/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py", line 3
058, in run_cell_async
      interactivity=interactivity, compiler=compiler, result=result)
    File "/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py", line 3
249, in run_ast_nodes
      if (await self.run_code(code, result,  async_=asy)):
    File "/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py", line 3
326, in run_code
      exec(code_obj, self.user_global_ns, self.user_ns)
    File "<ipython-input-41-13746e0edbe1>", line 1, in <module>
```

```
      historial = modelo.fit(Xtrain, Ytrain, epochs = 1000, verbose = False)
    File "/usr/local/lib/python3.7/dist-packages/keras/utils/traceback_utils.py", line 64,
in error_handler
      return fn(*args, **kwargs)
    File "/usr/local/lib/python3.7/dist-packages/keras/engine/training.py", line 1409, in
 fit
      tmp_logs = self.train_function(iterator)
    File "/usr/local/lib/python3.7/dist-packages/keras/engine/training.py", line 1051, in
 train_function
      return step_function(self, iterator)
    File "/usr/local/lib/python3.7/dist-packages/keras/engine/training.py", line 1040, in
 step_function
      outputs = model.distribute_strategy.run(run_step, args=(data,))
    File "/usr/local/lib/python3.7/dist-packages/keras/engine/training.py", line 1030, in
 run_step
      outputs = model.train_step(data)
    File "/usr/local/lib/python3.7/dist-packages/keras/engine/training.py", line 889, in t
rain_step
      y_pred = self(x, training=True)
    File "/usr/local/lib/python3.7/dist-packages/keras/utils/traceback_utils.py", line 64,
in error_handler
      return fn(*args, **kwargs)
    File "/usr/local/lib/python3.7/dist-packages/keras/engine/training.py", line 490, in _
_call__
      return super().__call__(*args, **kwargs)
    File "/usr/local/lib/python3.7/dist-packages/keras/utils/traceback_utils.py", line 64,
in error_handler
      return fn(*args, **kwargs)
    File "/usr/local/lib/python3.7/dist-packages/keras/engine/base_layer.py", line 1014, i
n __call__
      outputs = call_fn(inputs, *args, **kwargs)
    File "/usr/local/lib/python3.7/dist-packages/keras/utils/traceback_utils.py", line 92,
in error_handler
      return fn(*args, **kwargs)
    File "/usr/local/lib/python3.7/dist-packages/keras/engine/sequential.py", line 374, in
call
      return super(Sequential, self).call(inputs, training=training, mask=mask)
    File "/usr/local/lib/python3.7/dist-packages/keras/engine/functional.py", line 459, in
call
      inputs, training=training, mask=mask)
    File "/usr/local/lib/python3.7/dist-packages/keras/engine/functional.py", line 578, in
_run_internal_graph
      y = self._conform_to_reference_input(y, ref_input=x)
    File "/usr/local/lib/python3.7/dist-packages/keras/engine/functional.py", line 678, in
_conform_to_reference_input
      tensor = tf.cast(tensor, dtype=ref_input.dtype)
Node: 'sequential/Cast'
Cast string to float is not supported
         [[{{node sequential/Cast}}]] [Op:__inference_train_function_735]
```