125_exploracion_ANN

October 13, 2022

1 Exploración de modelos de red neuronal de PTA para vancomicina

1.1 Modelo de red neuronal para indicador primario

Indicador primario $AUC_{24}/MIC \geq 400$

[ ]:
```
import tensorflow as tf
from tensorflow import keras
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
```

[ ]:
```
home_dir = os.getcwd()
nets_dir = os.path.join(home_dir, '09_SIMULACION_QUIMIOLUMINISCENCIA', "model")
```

[ ]:
```
model = keras.models.load_model(os.path.join(nets_dir, "model_001"))
```

Para la predicción se debe utilizar una lista con los siguientes parámetros: `["Dosis Diaria", "Intervalo entre dosis", "Tiempo de infusión", "CRCL", "log2(MIC)"]`

Predicción para dosis de 2000 mg/d: 1g q12h tinf. 2 hrs

[ ]:
```
prediction_CRCL = lambda DD, CLCR: model.predict([[DD, 12, 2, CLCR, 0]],␣
 ↪verbose = 0)[0][0]
```

[ ]:
```
print(prediction_CRCL(2000, 90))
print(prediction_CRCL(2000, 100))
print(prediction_CRCL(2000, 120))
```

```
0.8542917
0.7918909
0.6174139
```

[ ]:
```
vec_CRCL = np.linspace(80, 170, 100)
vec_PTA  = [prediction_CRCL(2000, i) for i in vec_CRCL]
```

[ ]:

```python
def bisseccion(a, b, f, NA = 1e3, tol = 1e-6):
    i = 1
    fa = f(a)

    while i <= NA:
        p = a + (b - a)/2
        fp = f(p)

        if fp == 0 or (b - a)/2 < tol:
            return p
        i += 1

        if fa * fp > 0:
            a = p
            fa = fp
        else:
            b = p

    return None
```

[ ]:

```python
yintercept = bisseccion(80, 160, lambda x: prediction_CRCL(2000, x) - 0.85)
print(yintercept)
yintercept = bisseccion(80, 160, lambda x: prediction_CRCL(2000, x) - 0.90)
yintercept
```

```
90.71458876132965
```

[ ]:

```
80.36384999752045
```

[ ]:

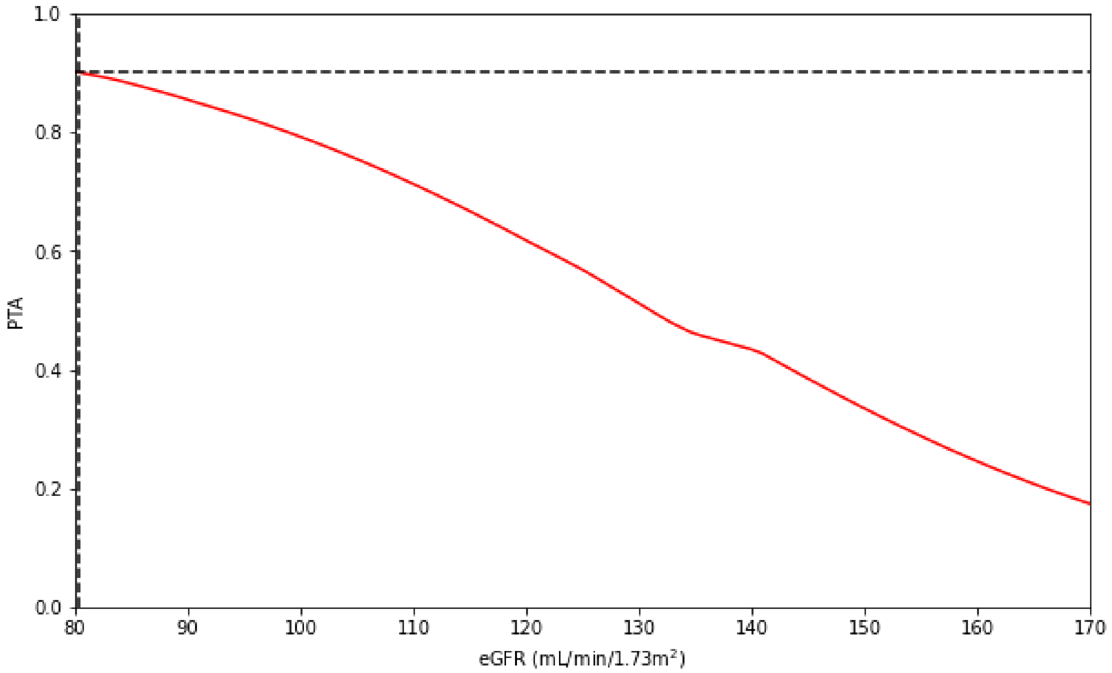
```python
fig, ax = plt.subplots(1,1, figsize = (10, 6))

ax.plot(vec_CRCL, vec_PTA, 'r-')
ax.axhline(0.90, color = 'k', linestyle = '--')
ax.axvline(yintercept, color = 'k', linestyle = '--')

ax.set_xlabel("eGFR (mL/min/1.73m$^2$)")
ax.set_ylabel("PTA")
ax.set_ylim(0, 1)
ax.set_xlim(80, 170)
```

[ ]:

```
(80.0, 170.0)
```



Predicción para dosis de 2500 mg: 1.25 g q12h tinf. 2 hrs

[ ]:

```
prediction_CRCL = lambda DD, CLCR: model.predict([[DD, 12, 2, CLCR, 0]],␣
  ↪verbose = 0)[0][0]
```

[ ]:

```
vec_CRCL = np.linspace(80, 170, 100)
vec_PTA  = [prediction_CRCL(2500, i) for i in vec_CRCL]
yintercept = bisseccion(90, 150, lambda x: prediction_CRCL(2500, x) - 0.85)
print(yintercept)
yintercept = bisseccion(90, 150, lambda x: prediction_CRCL(2500, x) - 0.90)
yintercept
```

```
132.26940125226974
```

[ ]:

```
122.43445247411728
```

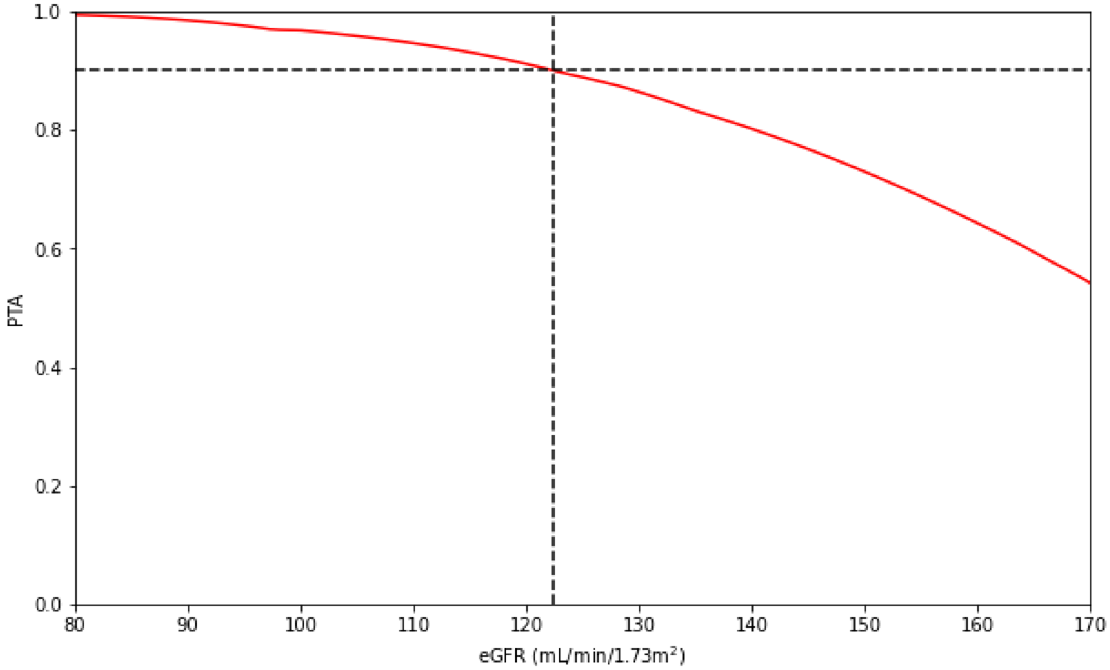
[ ]:

```
fig, ax = plt.subplots(1,1, figsize = (10, 6))

ax.plot(vec_CRCL, vec_PTA, 'r-')
ax.axhline(0.90, color = 'k', linestyle = '--')
ax.axvline(yintercept, color = 'k', linestyle = '--')

ax.set_xlabel("eGFR (mL/min/1.73m$^2$)")
ax.set_ylabel("PTA")
ax.set_ylim(0, 1)
ax.set_xlim(80, 170)
```

```
[ ]: (80.0, 170.0)
```



Predicción para dosis de 3000 mg: 1.25 g q12h tinf. 2 hrs

```
[ ]: vec_CRCL = np.linspace(80, 170, 100)
     vec_PTA  = [prediction_CRCL(3000, i) for i in vec_CRCL]
     yintercept = bisseccion(80, 170, lambda x: prediction_CRCL(3000, x) - 0.85)
     print(yintercept)
     yintercept = bisseccion(80, 170, lambda x: prediction_CRCL(3000, x) - 0.90)
     yintercept
```

```
166.6218952089548
```

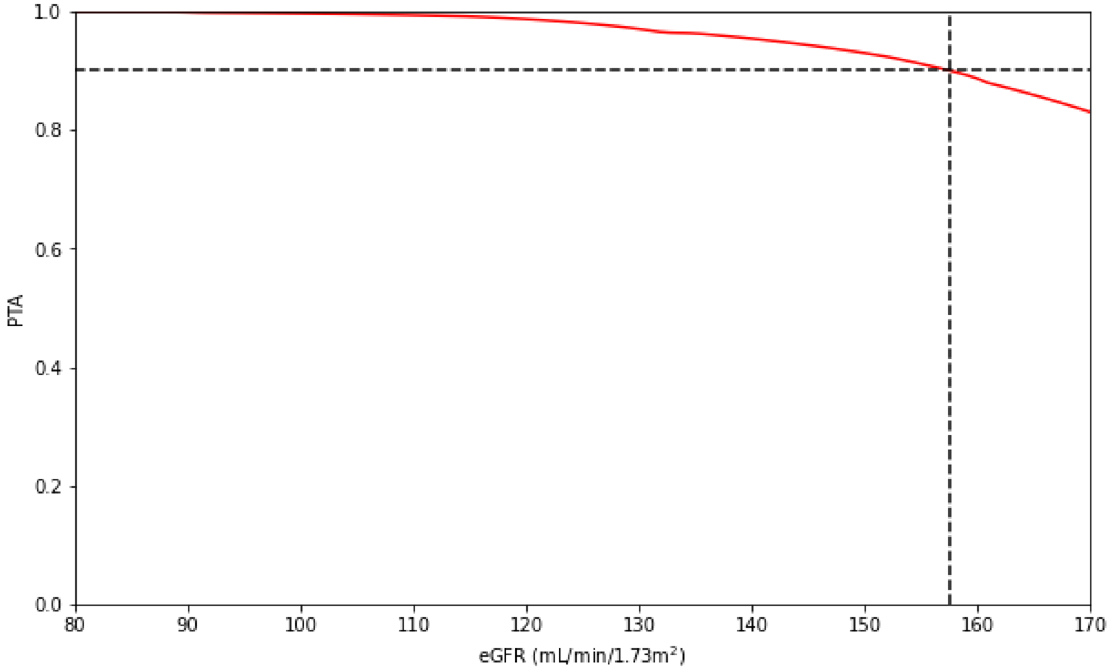
```
[ ]: 157.5102313607931
```

```
[ ]: fig, ax = plt.subplots(1,1, figsize = (10, 6))

     ax.plot(vec_CRCL, vec_PTA, 'r-')
     ax.axhline(0.90, color = 'k', linestyle = '--')
     ax.axvline(yintercept, color = 'k', linestyle = '--')

     ax.set_xlabel("eGFR (mL/min/1.73m$^2$)")
     ax.set_ylabel("PTA")
     ax.set_ylim(0, 1)
     ax.set_xlim(80, 170)
```

```
[ ]: (80.0, 170.0)
```


1.0
0.8
0.6
0.4
0.2
0.0
PTA
80
90
100
110
120
130
140
150
160
170
eGFR (mL/min/1.73m^2)

```
[ ]:
```