

Prática8_respostas

September 2, 2020

1 Prática 8

Aprendizado Dinâmico

por Cibele Russo (ICMC/USP - São Carlos SP)

MBA em Ciências de Dados

Nesta prática vamos considerar um modelo de sobrevivência com covariáveis, especificamente um modelo de Cox. Fonte: <https://www.kaggle.com/bryanb/survival-analysis-with-cox-model-implementation>

Deseja-se modelar o tempo até a morte de pacientes de câncer de mama em até 5 anos após uma cirurgia para a retirada de nódulos. Os dados são descritos no arquivo haberman.csv.

Descrição de dados

- Age: Idade do paciente no momento da operação;
- Operation_year: ano de operação do paciente (ano - 19XX);
- Nb_pos_detected: Número de nódulos axilares positivos detectados;
- Surv: Status de sobrevivência (atributo de classe) 1 = o paciente sobreviveu 5 anos ou mais
2 = o paciente morreu dentro de 5 anos;

Desenvolva um estudo com modelo de sobrevivência incluindo como covariáveis o ano da operação e o número de nódulos axilar positivos detectados.

1.Faça a leitura das bibliotecas.

```
[1]: import numpy as np
import pandas as pd
import pandas_profiling
import matplotlib.pyplot as plt

from lifelines import CoxPHFitter
from lifelines import KaplanMeierFitter

%matplotlib inline
```

2. Leia os dados do arquivo haberman.csv.

```
[2]: pkgdir = '/home/cibele/CibelePython/AprendizadoDinamico/Data'

data = pd.read_csv(f'{pkgdir}/haberman.csv', names = [
    → 'Age', 'Operation_year', 'Nb_pos_detected', 'Surv'])
data.head()
```

```
[2]:
```

	Age	Operation_year	Nb_pos_detected	Surv
0	30	64	1	1
1	30	62	3	1
2	30	65	0	1
3	31	59	2	1
4	31	65	4	1

3. Use o pandas_profiling para produzir um relatório de análises exploratórias

Comando:

```
pandas_profiling.ProfileReport(data)
```

```
[3]: pandas_profiling.ProfileReport(data)
```

```
HBox(children=(FloatProgress(value=0.0, description='Summarize dataset', max=18.0, style=ProgressStyle()),
```

```
HBox(children=(FloatProgress(value=0.0, description='Generate report structure', max=1.0, style=ProgressStyle()),
```

```
HBox(children=(FloatProgress(value=0.0, description='Render HTML', max=1.0, style=ProgressStyle()),
```

```
<IPython.core.display.HTML object>
```

```
[3]:
```

4. Estime a curva de sobrevivência com o estimador de Kaplan-Meier

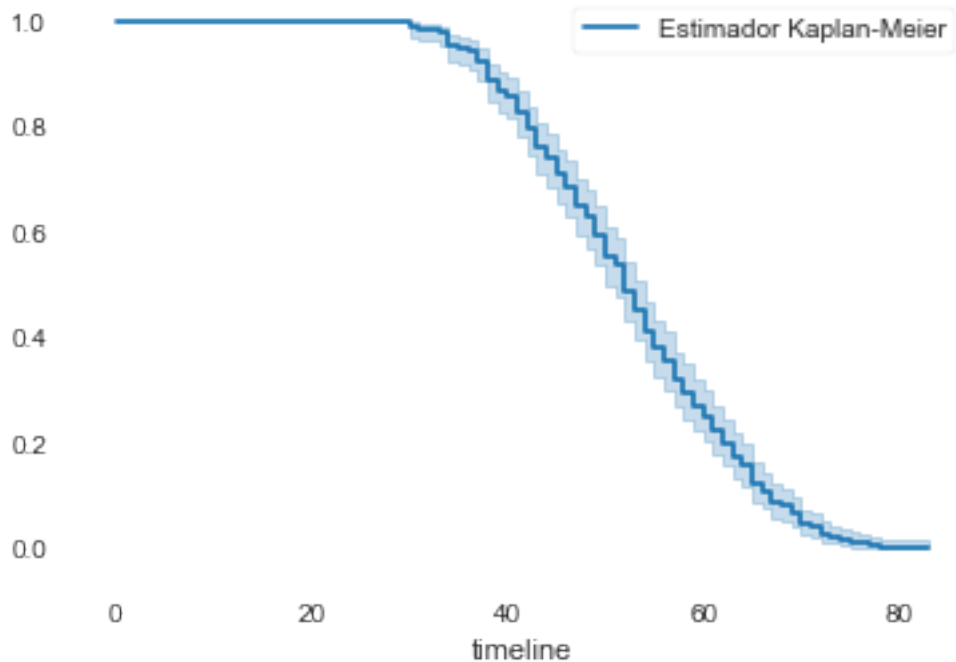
```
[4]: T = data.Age
E = data.Surv

km = KaplanMeierFitter()

km.fit(T, E, label='Estimador Kaplan-Meier')

km.plot(ci_show=True)
```

[4]: <AxesSubplot:xlabel='timeline'>



5. Utilize o estimador de Kaplan-Meier com dois grupos, um deles com pelo menos um nódulo axilar.

```
[5]: km_2 = KaplanMeierFitter()

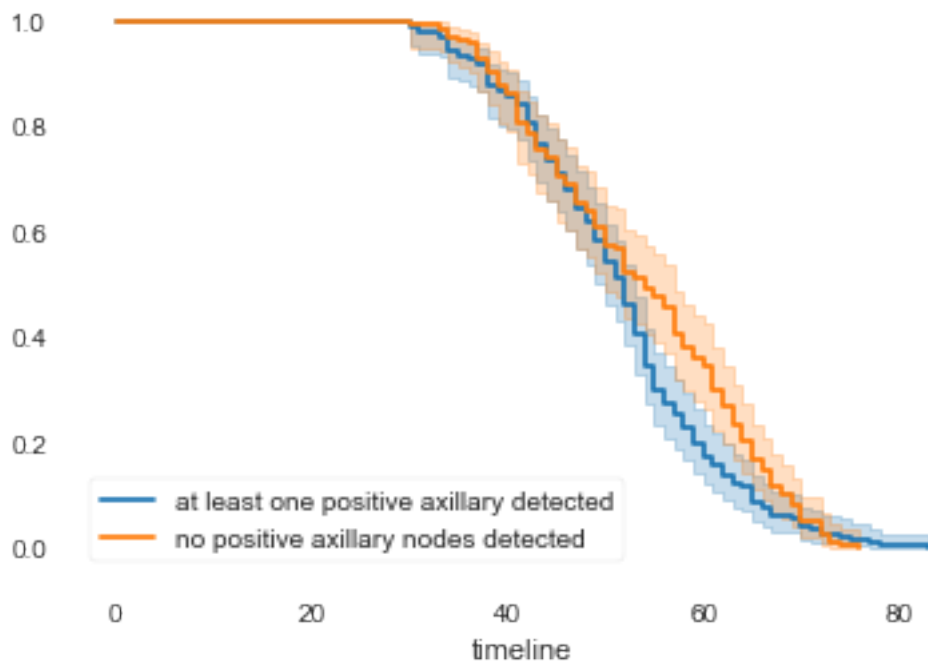
## Uma possibilidade: crie 2 coortes: com pelo menos um axilar positivo
→ detectado e um sem nenhum detectado

groups = data['Nb_pos_detected']
i1 = (groups >= 1)
i2 = (groups < 1)

## ajuste o modelo para a 1ª coorte
km_2.fit(T[i1], E[i1], label='at least one positive axillary detected')
a1 = km_2.plot()

## ajuste o modelo para a 2ª coorte
km_2.fit(T[i2], E[i2], label='no positive axillary nodes detected')
km_2.plot(ax=a1)
```

[5]: <AxesSubplot:xlabel='timeline'>



6. Ajuste um modelo de Cox com covariáveis Operatio_year e Nb_pos_detected. Interprete os resultados

```
[6]: # Crie o modelo
cph = CoxPHFitter()

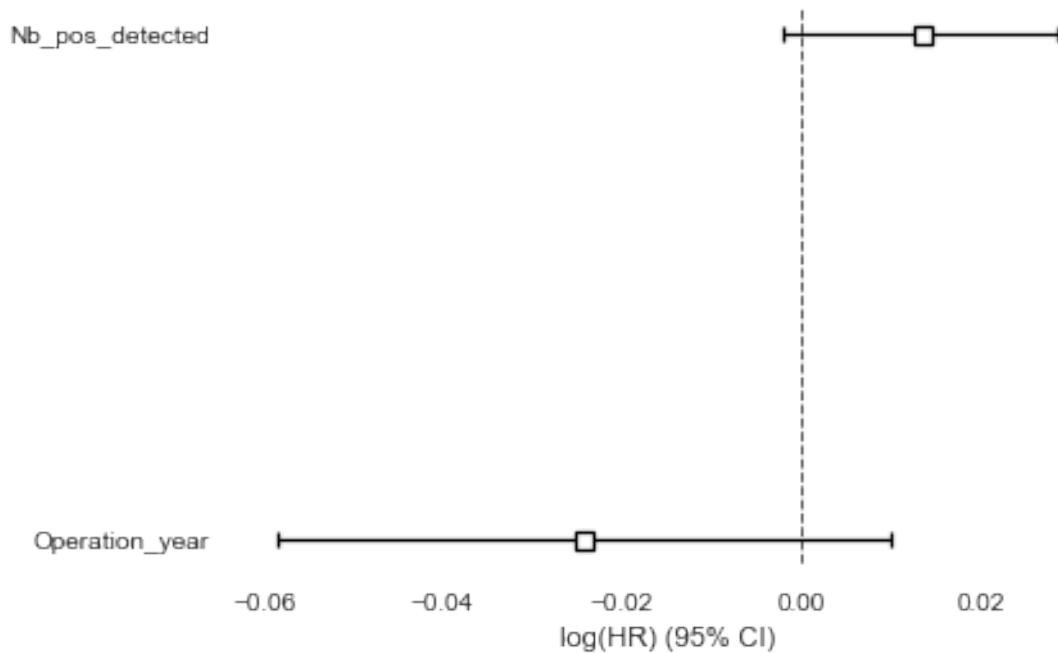
# Ajuste o modelo
cph.fit(data, 'Age', event_col='Surv')

# Verifique os resultados
cph.print_summary()
```

covariate	coef	exp(coef)	se(coef)	coef lower 95%	coef upper 95%	exp(coef) lower 95%	exp(coef) upper 95%
Operation_year	-0.02	0.98	0.02	-0.06	0.01	0.94	1.00
Nb_pos_detected	0.01	1.01	0.01	-0.00	0.03	0.99	1.03

```
[7]: cph.plot()
```

```
[7]: <AxesSubplot:xlabel='log(HR) (95% CI)'\>
```



7. Selecione três pacientes de forma aleatória para fazer a previsão da curva de sobrevivência para cada um deles

```
[8]: patients = [4,125,211]

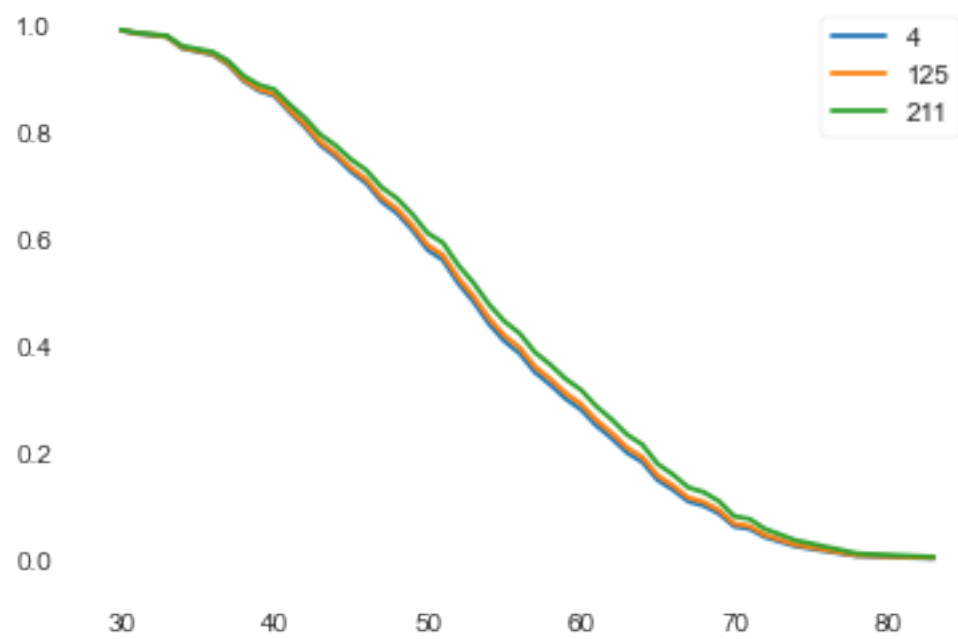
rows_selected = data.iloc[patients, 1:3]
rows_selected
```

```
[8]:      Operation_year  Nb_pos_detected
4              65              4
125             64              0
211             67              0
```

8. Faça a previsão da curva de sobrevivência para cada um dos pacientes selecionados acima

```
[9]: cph.predict_survival_function(rows_selected).plot()
```

```
[9]: <AxesSubplot:>
```



[]: