EDA_Haberman_Dataset

December 15, 2018

0.1 EDA - Haberman's Survival Data

0.2 Assignment description

- Download Haberman Cancer Survival dataset from Kaggle. You may have to create a Kaggle account to donwload data. (https://www.kaggle.com/gilsousa/habermans-survival-data-set)
- Perform a similar analysis as above on this dataset with the following sections:
- High level statistics of the dataset: number of points, numer of features, number of classes, data-points per class.
- Explain our objective.
- Perform Univaraite analysis(PDF, CDF, Boxplot, Voilin plots) to understand which features are useful towards classification.
- Perform Bi-variate analysis (scatter plots, pair-plots) to see if combinations of features are useful in classfication.
- Write your observations in english as crisply and unambigously as possible. Always quantify your results.

0.3 Dataset qualitive description

Dataset source: (a) Donor: Tjen-Sien Lim (limt@stat.wisc.edu) (b) Date: March 4, 1999

Features 1. Age of patient at time of operation (numerical) 2. Patient's year of operation (year - 1900, numerical) 3. Number of positive axillary nodes detected (numerical)

Axillary nodes drain lymph vessels from the lateral quadrants of the breast, and are clinically significant in breast cancer.

Classes * 1 (survived) = the patient survived 5 years or longer * 2 (died) = the patient died within 5 year

```
In [5]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns

In [6]: haberman = pd.read_csv('./haberman.csv',
                                          names = ["age", "operation_year", "axillary_nodes", "surviva

        # preprocessing to facilitate the analysis
        haberman.loc[haberman.survival_status == 1, 'survival_status'] = 'survived'
        haberman.loc[haberman.survival_status == 2, 'survival_status'] = 'died'
```

```
In [7]: # first dataset visualization
        print(haberman)
        print(haberman.describe())
```

```
     age  operation_year  axillary_nodes survival_status
0     30              64               1        survived
1     30              62               3        survived
2     30              65               0        survived
3     31              59               2        survived
4     31              65               4        survived
5     33              58              10        survived
6     33              60               0        survived
7     34              59               0            died
8     34              66               9            died
9     34              58              30        survived
10    34              60               1        survived
11    34              61              10        survived
12    34              67               7        survived
13    34              60               0        survived
14    35              64              13        survived
15    35              63               0        survived
16    36              60               1        survived
17    36              69               0        survived
18    37              60               0        survived
19    37              63               0        survived
20    37              58               0        survived
21    37              59               6        survived
22    37              60              15        survived
23    37              63               0        survived
24    38              69              21            died
25    38              59               2        survived
26    38              60               0        survived
27    38              60               0        survived
28    38              62               3        survived
29    38              64               1        survived
..   ...             ...             ...             ...
276   67              66               0        survived
277   67              61               0        survived
278   67              65               0        survived
279   68              67               0        survived
280   68              68               0        survived
281   69              67               8            died
282   69              60               0        survived
283   69              65               0        survived
284   69              66               0        survived
285   70              58               0            died
286   70              58               4            died
287   70              66              14        survived
```

```
288   70              67               0        survived
289   70              68               0        survived
290   70              59               8        survived
291   70              63               0        survived
292   71              68               2        survived
293   72              63               0            died
294   72              58               0        survived
295   72              64               0        survived
296   72              67               3        survived
297   73              62               0        survived
298   73              68               0        survived
299   74              65               3            died
300   74              63               0        survived
301   75              62               1        survived
302   76              67               0        survived
303   77              65               3        survived
304   78              65               1            died
305   83              58               2            died

[306 rows x 4 columns]
              age  operation_year  axillary_nodes
count  306.000000      306.000000      306.000000
mean    52.457516       62.852941        4.026144
std     10.803452        3.249405        7.189654
min     30.000000       58.000000        0.000000
25%     44.000000       60.000000        0.000000
50%     52.000000       63.000000        1.000000
75%     60.750000       65.750000        4.000000
max     83.000000       69.000000       52.000000
```

0.3.1 1 High level statistics of the dataset

```
In [8]: print(haberman.columns)
        print(haberman.shape)
        # number of points: 306
        # number of features: 3
        # number of classes: 2

        gb = haberman.groupby('survival_status')
        print(gb.count())

        # data-points per class:
        # 1 (the patient survived 5 years or longer): 225
        # 2 (the patient died within 5 year): 81
```

```
Index(['age', 'operation_year', 'axillary_nodes', 'survival_status'], dtype='object')
(306, 4)
```

```
                 age  operation_year  axillary_nodes
survival_status
died              81              81              81
survived         225             225             225
```

0.3.2 2 Objective

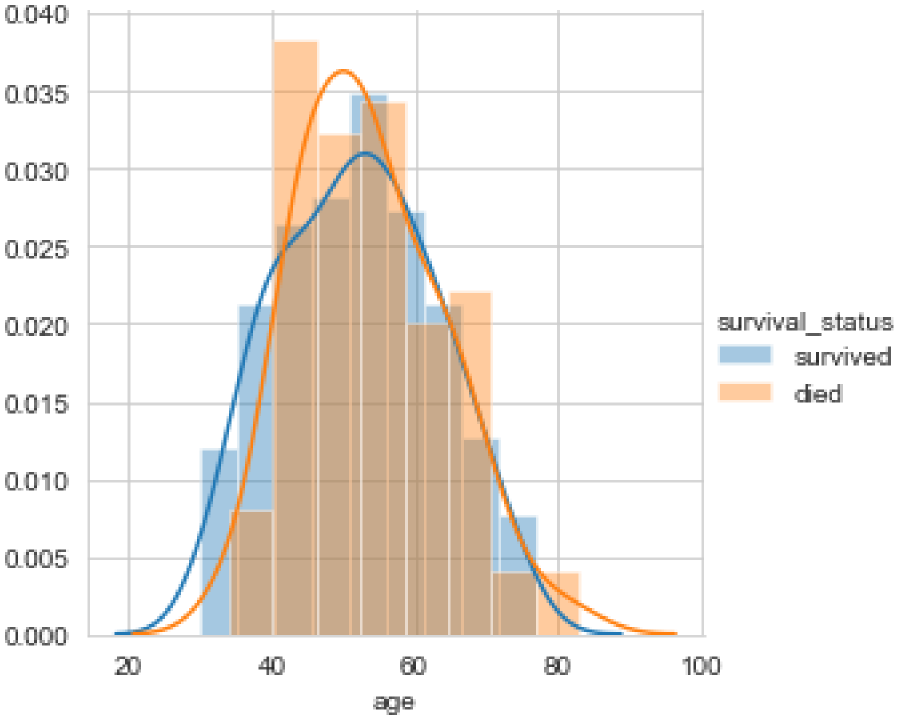
Classify a new patient that did a surgery for breast cancer as belonging to one of the 2 classes, given the 3 features described in the "Dataset qualitive description" section.

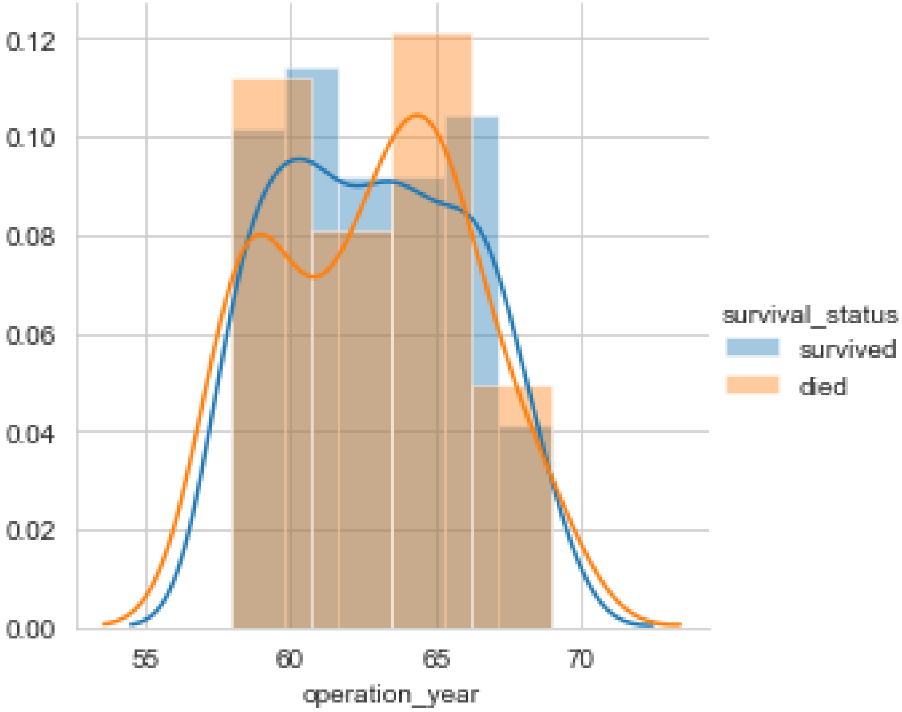
0.3.3 3 Univariate analysis

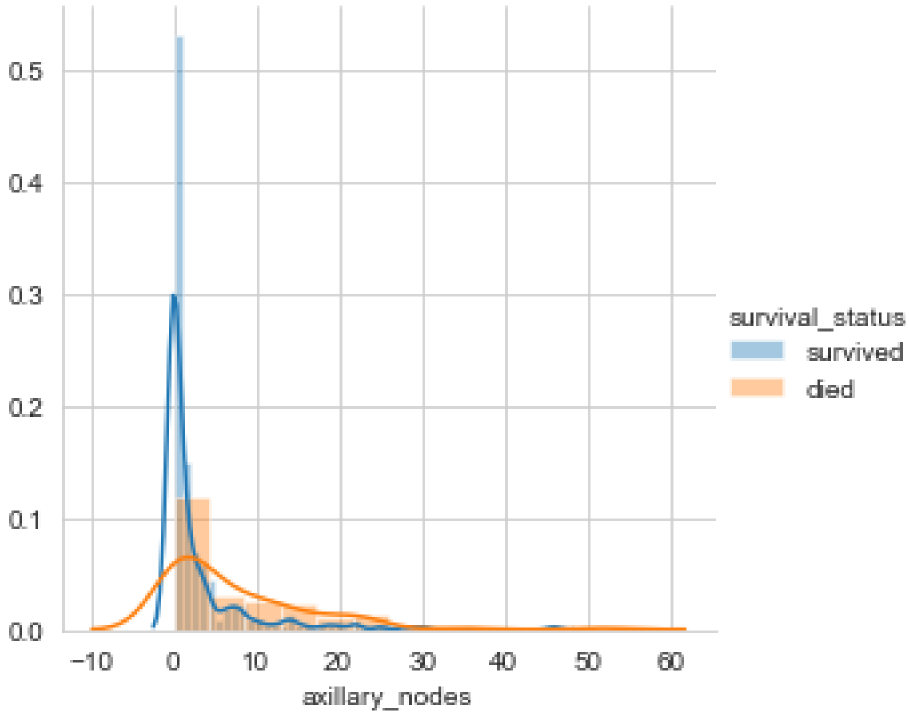
3.1 PDF

```
In [9]: sns.set_style('whitegrid')

        sns.FacetGrid(haberman, hue='survival_status', height=4) \
                     .map(sns.distplot, 'age') \
                     .add_legend()

        sns.FacetGrid(haberman, hue='survival_status', height=4) \
                     .map(sns.distplot, 'operation_year') \
                     .add_legend()


        sns.FacetGrid(haberman, hue='survival_status', height=4) \
                     .map(sns.distplot, 'axillary_nodes') \
                     .add_legend()

        plt.show()
```

```
/Users/gustavo.fonseca/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureW
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```


0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
20
40
60
80
100
age
survival_status
survived
died


0.12
0.10
0.08
0.06
0.04
0.02
0.00
55
60
65
70
operation_year
survival_status
survived
died



3.1.1 Observation(s) Age and Operation_year graphs: * They both assume a normal distribution; * Both of their classes overlap tremendously, therefore, they do not appear to be separable.

Axillary_nodes graph: * It assumes a right skewed form; * Both classes overlap; * The PDF of the survived class is much higher in approx 0 axillary_nodes.

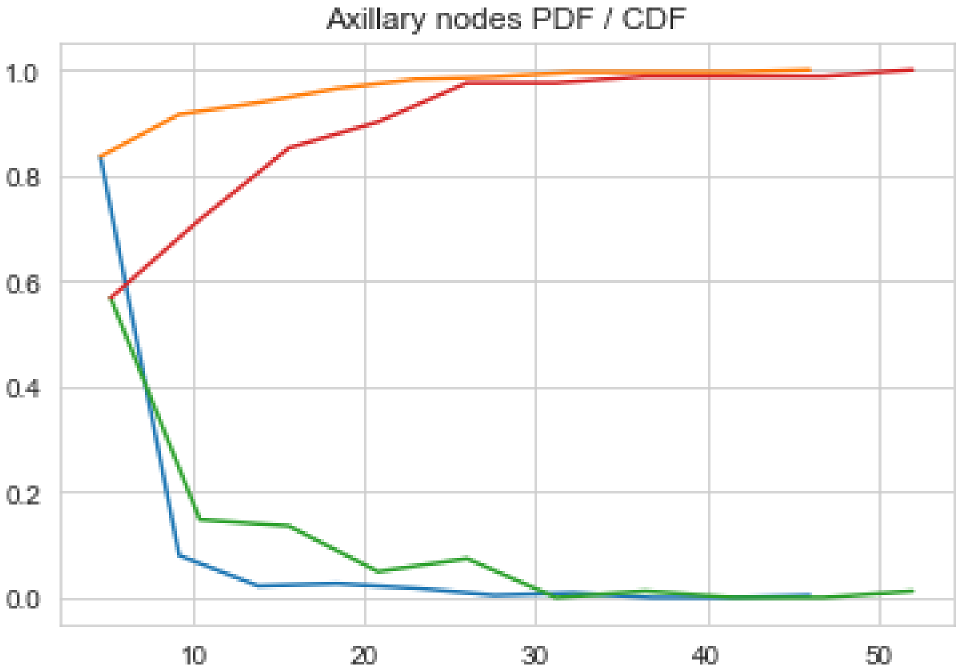
3.2 CDF

Based on the PDF plot, age and operation_year do not appear as promising for univariate analysis as axillary_nodes. Hence, only the axillary nodes' CDF is plotted for further visualization and analysis.

```
In [10]: # divide the classes to perform separate analysis
         haberman_survived = haberman.loc[haberman["survival_status"] == "survived"];
         haberman_died = haberman.loc[haberman["survival_status"] == "died"];
```

```
In [11]: plt.figure(1)
         plt.title('Axillary nodes PDF / CDF')

         # survived
         counts, bin_edges = np.histogram(haberman_survived.axillary_nodes, bins=10,
                                          density = True)
         pdf = counts/(sum(counts))
         cdf = np.cumsum(pdf)
         plt.plot(bin_edges[1:], pdf);
         plt.plot(bin_edges[1:], cdf)
```

```
# died
counts, bin_edges = np.histogram(haberman_died.axillary_nodes, bins=10,
                                 density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:], pdf);
plt.plot(bin_edges[1:], cdf)

plt.show()
```



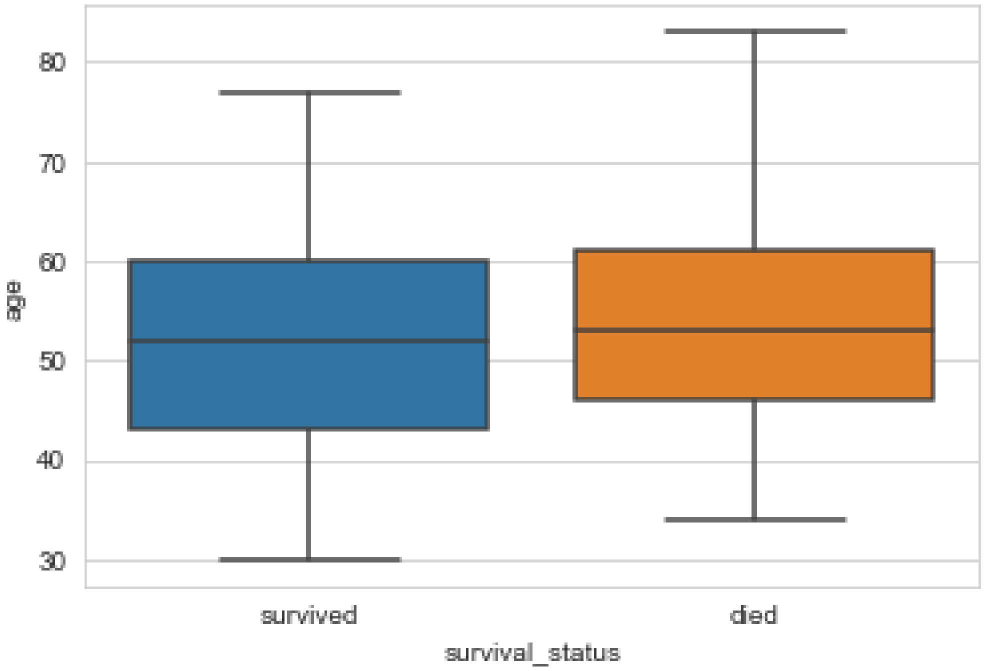
3.2.2 Observations

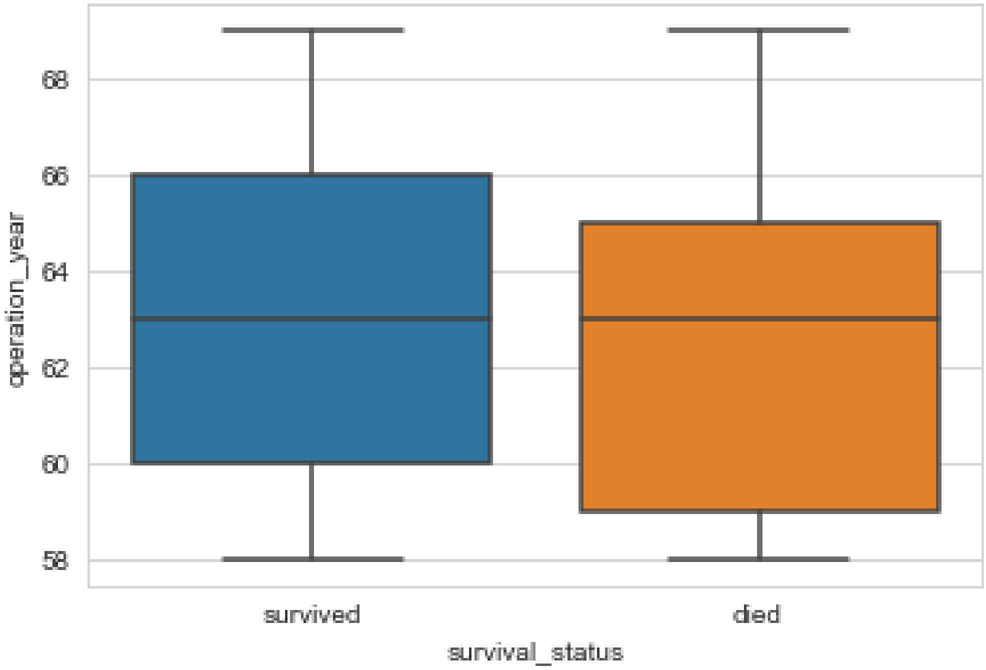
- At the 10 axillary nodes removal lies: approx 90% of all survivals and 70% of all non-survivals;
- With only one feature, it doens't seem to exist a basic model capable of separating both classes efficiently.

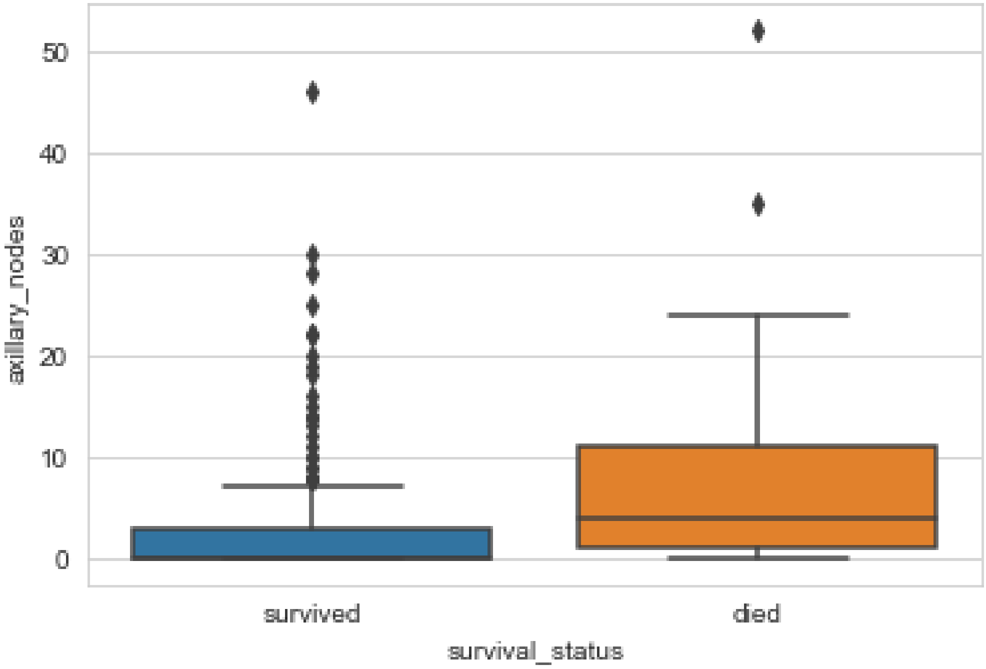
3.3 Box and Whiskers

```
In [12]: plt.close()

         plt.figure(1)
         sns.boxplot(x='survival_status', y='age', data=haberman)
```

```
plt.figure(2)
sns.boxplot(x='survival_status', y='operation_year', data=haberman)

plt.figure(3)
sns.boxplot(x='survival_status', y='axillary_nodes', data=haberman)

plt.show()
```




68
66
64
62
60
58
operation_year
survived
died
survival_status


50
40
30
20
10
0
axillary_nodes
survived
died
survival_status

3.3.1 Observations Age graph: * Both classes have a very similiar median, ranging between 52 - 54; * Even though both classes' boxes are in a similiar range, older patients seem to die more frequently.

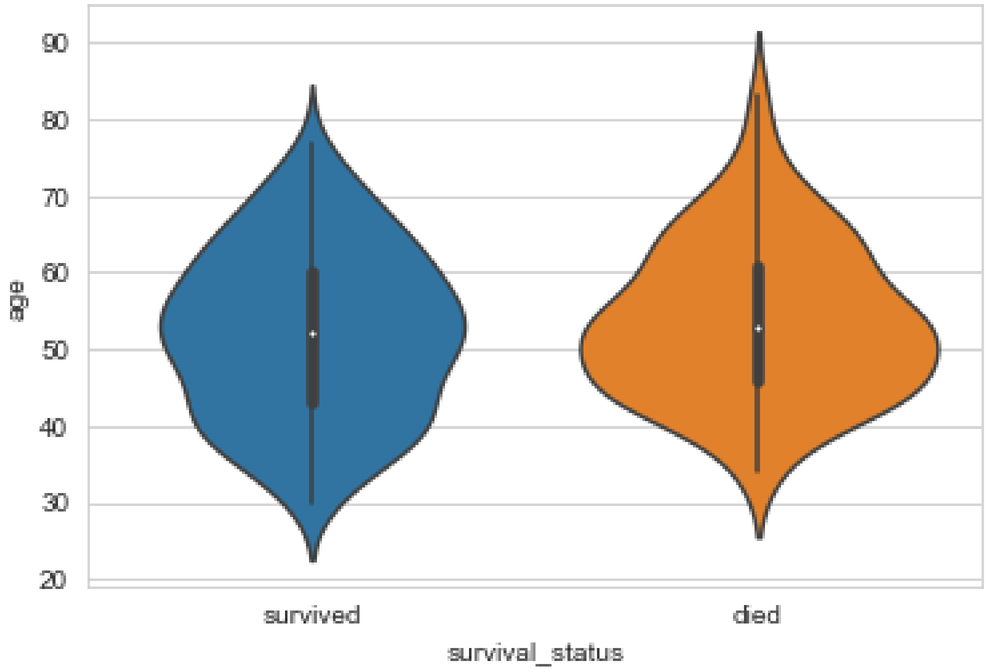
Operation year: * Both classes seem to have an identical median, approx 63; * Both classes lies in a similar range, but patients who have an older operation year, seem to die more frequently.

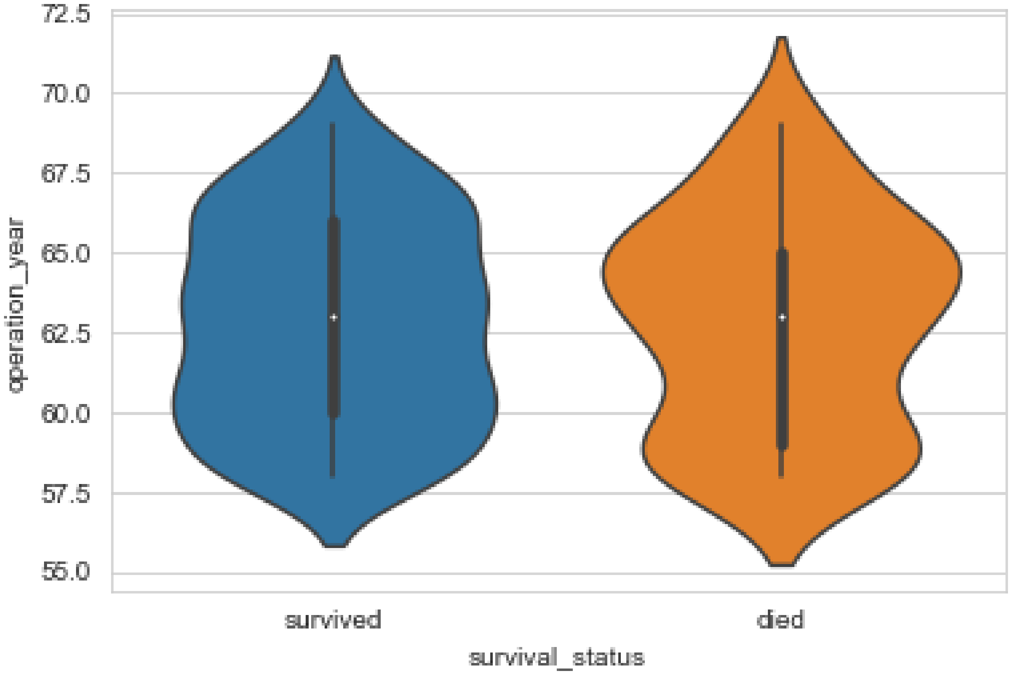
Axillary nodes: * There are a significant number of outliers; * It appears to be the most promising feature to reach the objective of this model; * Patients that remove less axillary_nodes appears to have a higher survival chance;

3.4 Violin

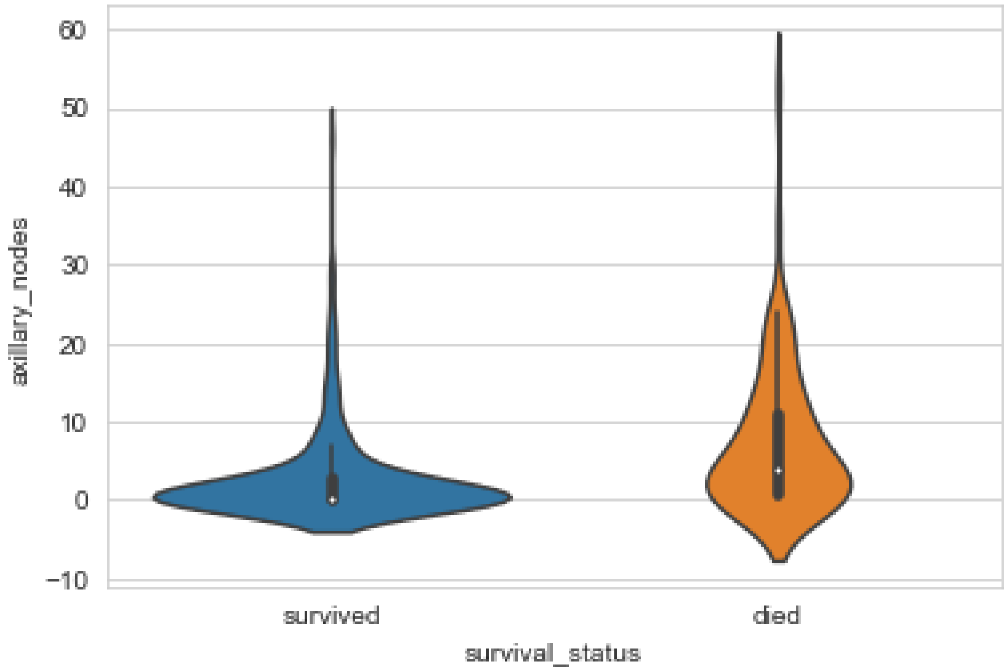
```
In [13]: plt.close()

         plt.figure(1)
         sns.violinplot(x="survival_status", y="age", data=haberman, size=8)

         plt.figure(2)
         sns.violinplot(x="survival_status", y="operation_year", data=haberman, size=8)

         plt.figure(3)
         sns.violinplot(x="survival_status", y="axillary_nodes", data=haberman, size=8)

         plt.show()
```

```
/Users/gustavo.fonseca/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureW
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```





72.5
70.0
67.5
65.0
62.5
60.0
57.5
55.0
operation_year
survived
died
survival_status



60
50
40
30
20
10
0
−10
axillary_nodes
survived
died
survival_status

3.4.1 Observations

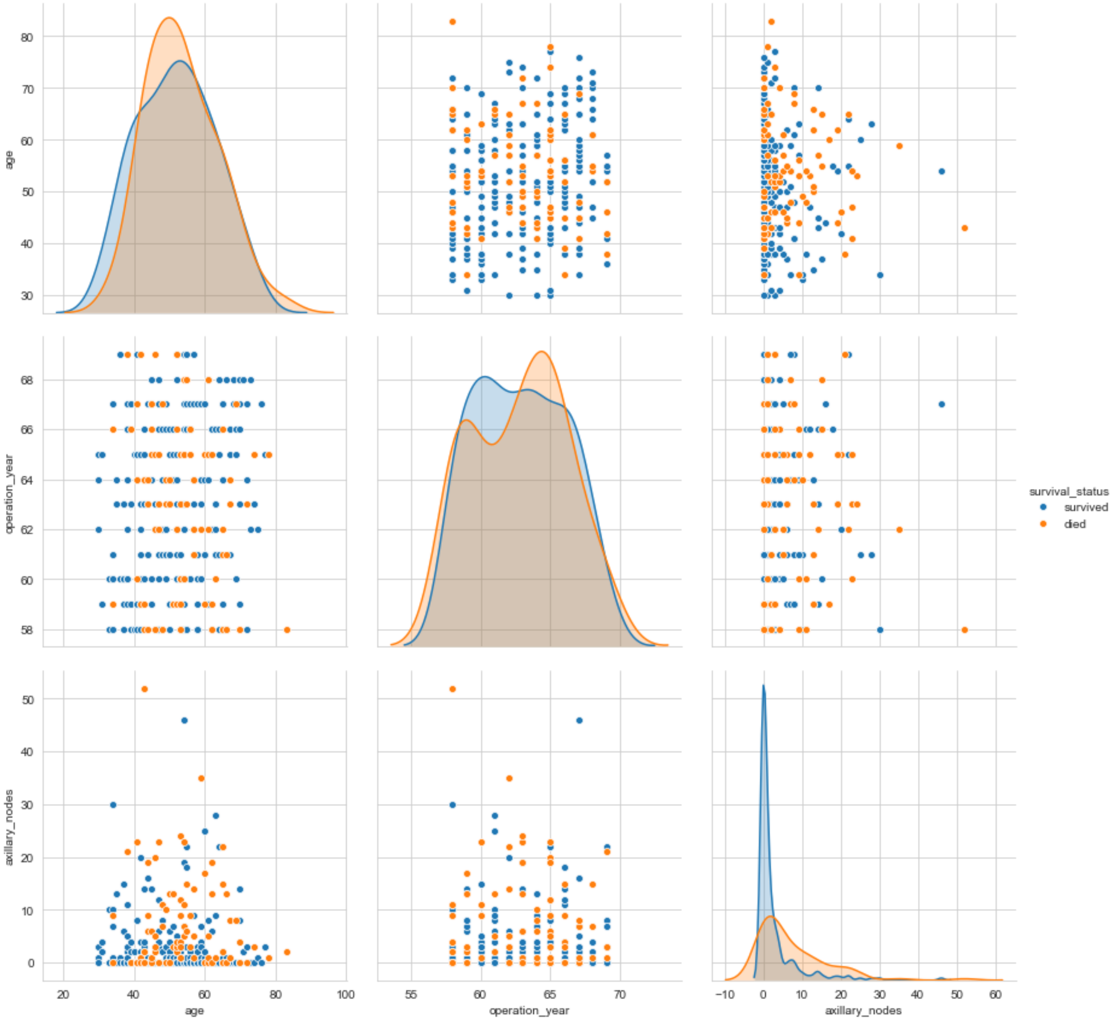
- The violin plot shows that patients that remove 0 axillary nodes have a high probability of survival.

0.3.4 4. Bivariate analysis

4.1 Pair plots

```
In [14]: plt.close();
         sns.set_style("whitegrid");
         sns.pairplot(haberman, hue="survival_status", height=4);
         plt.show()
```

```
/Users/gustavo.fonseca/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureW
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



4. 2 Observations

- Apparently there is no pair of features that could linearly separate both classes effectively;
- The most promising feature for classification seems to be axillary_nodes;
- We can't build a simple model with 'if' and 'else' to separate both classes efficiently.