c-machine-learning-from-disaster

February 3, 2024

1 Titanic - Machine Learning from Disaster

[1]:
```
import numpy as np
import pandas as pd
```

[75]:
```
df = pd.read_csv('test.csv')
```

[76]:
```
df.head()
```

[76]:
```
   PassengerId  Pclass                                          Name     Sex  \
0          892       3                              Kelly, Mr. James    male
1          893       3              Wilkes, Mrs. James (Ellen Needs)  female
2          894       2                     Myles, Mr. Thomas Francis    male
3          895       3                              Wirz, Mr. Albert    male
4          896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female

    Age  SibSp  Parch   Ticket     Fare Cabin Embarked
0  34.5      0      0   330911   7.8292   NaN        Q
1  47.0      1      0   363272   7.0000   NaN        S
2  62.0      0      0   240276   9.6875   NaN        Q
3  27.0      0      0   315154   8.6625   NaN        S
4  22.0      1      1  3101298  12.2875   NaN        S
```

[6]:
```
df.isnull().sum()
```

[6]:
```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

```
[77]: x = df.drop(columns=['Sex'])
```

```
[79]: y= df['Sex']
```

```
[81]: df.Sex = df.Sex.map({'male':0 , 'female':1})
```

```
[82]: df
```

```
[82]:      PassengerId  Pclass                                          Name  Sex  \
      0            892       3                              Kelly, Mr. James    0
      1            893       3              Wilkes, Mrs. James (Ellen Needs)    1
      2            894       2                     Myles, Mr. Thomas Francis    0
      3            895       3                              Wirz, Mr. Albert    0
      4            896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)    1
      ..           …     …                                            …   …
      413         1305       3                            Spector, Mr. Woolf    0
      414         1306       1                  Oliva y Ocana, Dona. Fermina    1
      415         1307       3                  Saether, Mr. Simon Sivertsen    0
      416         1308       3                           Ware, Mr. Frederick    0
      417         1309       3                      Peter, Master. Michael J    0

            Age  SibSp  Parch              Ticket      Fare Cabin Embarked
      0    34.5      0      0              330911    7.8292   NaN        Q
      1    47.0      1      0              363272    7.0000   NaN        S
      2    62.0      0      0              240276    9.6875   NaN        Q
      3    27.0      0      0              315154    8.6625   NaN        S
      4    22.0      1      1             3101298   12.2875   NaN        S
      ..    …    …    …                 …     …    …     …
      413   NaN      0      0          A.5. 3236    8.0500   NaN        S
      414  39.0      0      0            PC 17758  108.9000  C105        C
      415  38.5      0      0  SOTON/O.Q. 3101262    7.2500   NaN        S
      416   NaN      0      0              359309    8.0500   NaN        S
      417   NaN      1      1                2668   22.3583   NaN        C

      [418 rows x 11 columns]
```

```
[83]: y = df['Sex']
```

```
[84]: y
```

```
[84]: 0      0
      1      1
      2      0
      3      0
      4      1
            ..
      413    0
```

```
414    1
415    0
416    0
417    0
Name: Sex, Length: 418, dtype: int64
```

[9]:
```
df.duplicated().sum()
```

[9]:
```
0
```

[11]:
```
df.corr()
```

```
/var/folders/8c/20t35gwd03j9m2lldclbwr6c0000gn/T/ipykernel_75777/1134722465.py:1
: FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid
columns or specify the value of numeric_only to silence this warning.
  df.corr()
```

[11]:
```
             PassengerId    Pclass       Age     SibSp     Parch      Fare
PassengerId     1.000000 -0.026751 -0.034102  0.003818  0.043080  0.008211
Pclass         -0.026751  1.000000 -0.492143  0.001087  0.018721 -0.577147
Age            -0.034102 -0.492143  1.000000 -0.091587 -0.061249  0.337932
SibSp           0.003818  0.001087 -0.091587  1.000000  0.306895  0.171539
Parch           0.043080  0.018721 -0.061249  0.306895  1.000000  0.230046
Fare            0.008211 -0.577147  0.337932  0.171539  0.230046  1.000000
```

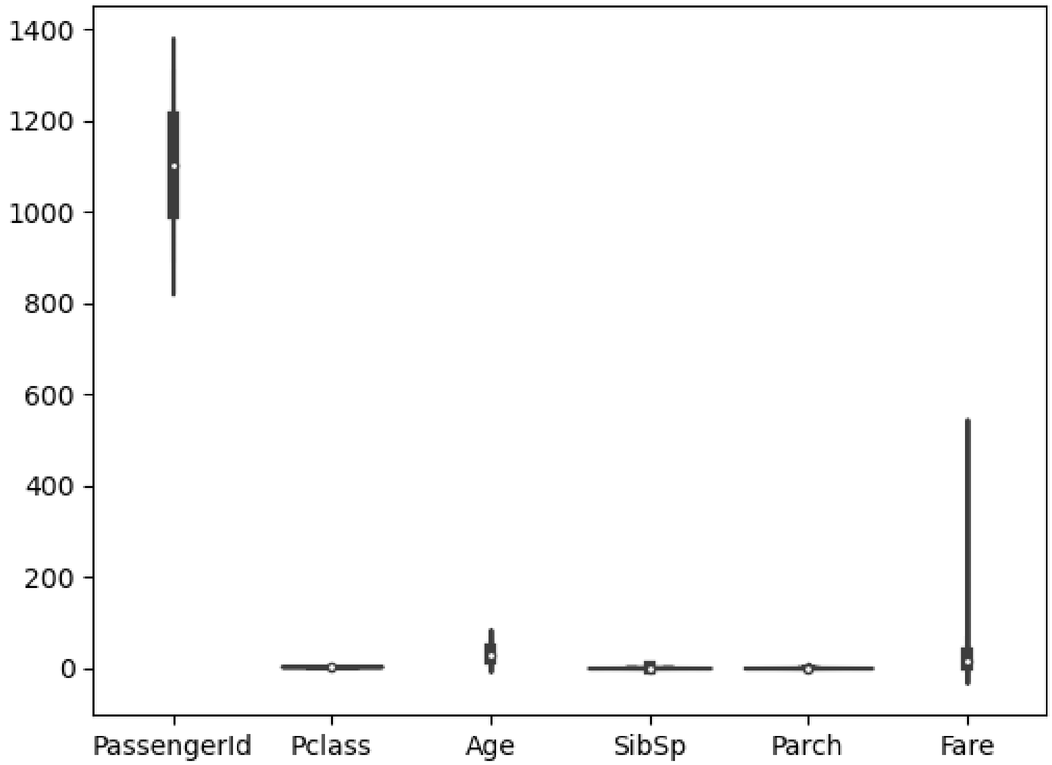
[12]:
```
df.dtypes
```

[12]:
```
PassengerId      int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

[13]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

[14]:
```
sns.violinplot(data=df)
```

[14]:
```
<Axes: >
```



[15]:
```
df.head()
```

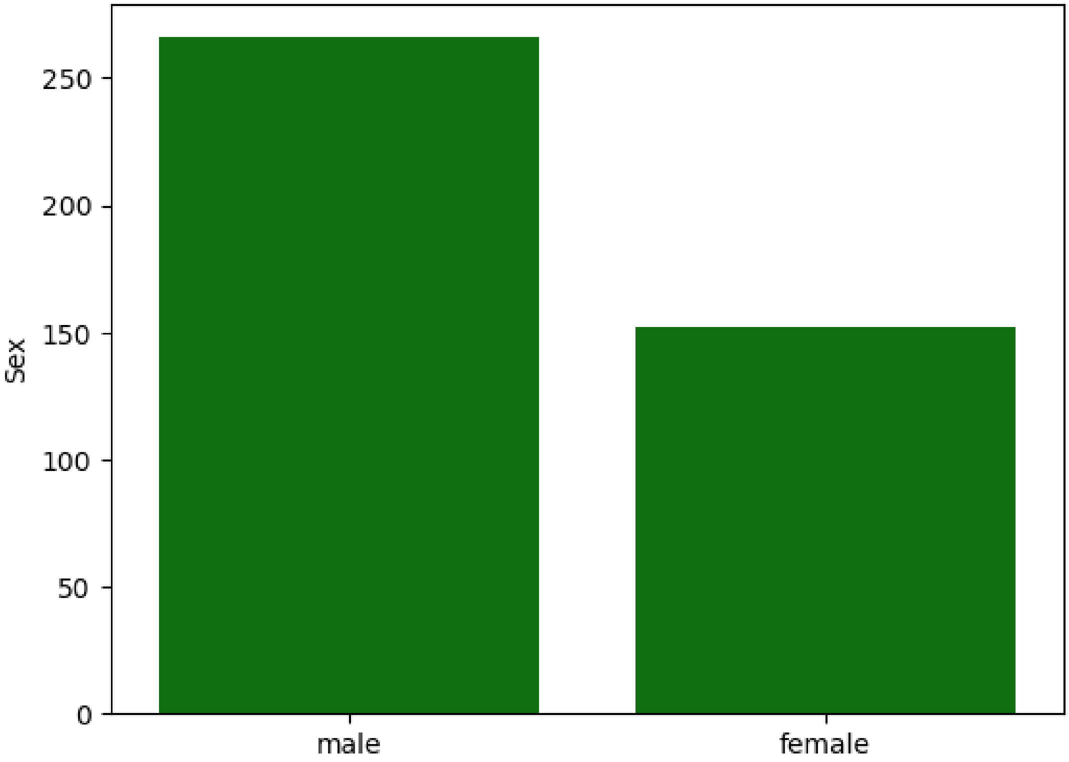
[15]:
```
   PassengerId  Pclass                                          Name     Sex  \
0          892       3                              Kelly, Mr. James    male
1          893       3              Wilkes, Mrs. James (Ellen Needs)  female
2          894       2                     Myles, Mr. Thomas Francis    male
3          895       3                              Wirz, Mr. Albert    male
4          896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female

    Age  SibSp  Parch   Ticket     Fare Cabin Embarked
0  34.5      0      0   330911   7.8292   NaN        Q
1  47.0      1      0   363272   7.0000   NaN        S
2  62.0      0      0   240276   9.6875   NaN        Q
3  27.0      0      0   315154   8.6625   NaN        S
4  22.0      1      1  3101298  12.2875   NaN        S
```

[16]:
```
df['Sex'].value_counts()
```

[16]:
```
male      266
female    152
Name: Sex, dtype: int64
```

[18]:
```
sns.barplot(x=df['Sex'].unique() , y=df['Sex'].value_counts(), color='green')
plt.show()
```



[19]:
```
df.head(2)
```

[19]:
```
   PassengerId  Pclass                                Name     Sex   Age  SibSp  \
0          892       3                    Kelly, Mr. James    male  34.5      0
1          893       3  Wilkes, Mrs. James (Ellen Needs)  female  47.0      1

   Parch  Ticket    Fare Cabin Embarked
0      0  330911  7.8292   NaN        Q
1      0  363272  7.0000   NaN        S
```

[20]:
```
df.Sex = df.Sex.map({'male':0 , 'female':1})
```

[72]:
```
df
```

[72]:
```
     Pclass  Name  Age  SibSp  Parch  Ticket  Fare  Embarked
0         2   206   44      0      0     152    24         1
1         2   403   60      1      0     221     5         2
2         1   269   74      0      0      73    41         1
3         2   408   34      0      0     147    34         2
```

```
4         2   178   27      1      1     138    46         2
..      …   …  …    …    …     …   …       …
413       2   353   79      0      0     267    31         2
414       0   283   51      0      0     324   154         0
415       2   332   50      0      0     346     9         2
416       2   384   79      0      0     220    31         2
417       2   302   79      1      1     105    84         0

[418 rows x 8 columns]
```

[42]:

```
df.drop(columns=['Cabin'], inplace=True)
```

[44]:

```
df.drop(columns=['PassengerId'], inplace=True)
```

[54]:

```
df
```

[54]:

```
     Pclass                                           Name   Age  SibSp  Parch  \
0         3                               Kelly, Mr. James  34.5      0      0
1         3               Wilkes, Mrs. James (Ellen Needs)  47.0      1      0
2         2                      Myles, Mr. Thomas Francis  62.0      0      0
3         3                               Wirz, Mr. Albert  27.0      0      0
4         3   Hirvonen, Mrs. Alexander (Helga E Lindqvist)  22.0      1      1
..      …                                            …   …    …    …
413       3                             Spector, Mr. Woolf   NaN      0      0
414       1                   Oliva y Ocana, Dona. Fermina  39.0      0      0
415       3                   Saether, Mr. Simon Sivertsen  38.5      0      0
416       3                            Ware, Mr. Frederick   NaN      0      0
417       3                       Peter, Master. Michael J   NaN      1      1

                 Ticket      Fare Embarked
0                330911    7.8292        Q
1                363272    7.0000        S
2                240276    9.6875        Q
3                315154    8.6625        S
4               3101298   12.2875        S
..                  …       …      …
413           A.5. 3236    8.0500        S
414            PC 17758  108.9000        C
415  SOTON/O.Q. 3101262    7.2500        S
416              359309    8.0500        S
417                2668   22.3583        C

[418 rows x 8 columns]
```

[61]:

```
col_list = list(df.columns)
```

[59]:

```
col_list
```

```
[59]: ['Pclass', 'Name', 'Age', 'SibSp', 'Parch', 'Ticket', 'Fare', 'Embarked']
```

```
[62]: col_list = df.columns.tolist()
```

```
[63]: for col in col_list:
          print(col)
```

```
Pclass
Name
Age
SibSp
Parch
Ticket
Fare
Embarked
```

```
[64]: from sklearn.preprocessing import LabelEncoder
```

```
[65]: label_encoder = LabelEncoder()
```

```
[91]: columns = ['Pclass', 'Name', 'Age', 'SibSp', 'Parch', 'Ticket', 'Fare',␣
        ↪'Embarked']
      df[columns] = df[columns].apply(lambda col: label_encoder.fit_transform(col))
```

```
[92]: columns
```

```
[92]: ['Pclass', 'Name', 'Age', 'SibSp', 'Parch', 'Ticket', 'Fare', 'Embarked']
```

```
[93]: col_list
```

```
[93]: ['Pclass', 'Name', 'Age', 'SibSp', 'Parch', 'Ticket', 'Fare', 'Embarked']
```

```
[94]: df
```

```
[94]:      PassengerId  Pclass  Name  Sex  Age  SibSp  Parch  Ticket  Fare Cabin  \
      0            892       2   206    0   44      0      0     152    24   NaN
      1            893       2   403    1   60      1      0     221     5   NaN
      2            894       1   269    0   74      0      0      73    41   NaN
      3            895       2   408    0   34      0      0     147    34   NaN
      4            896       2   178    1   27      1      1     138    46   NaN
      ..           …      …    …   …  …     …    …      …   …    …
      413         1305       2   353    0   79      0      0     267    31   NaN
      414         1306       0   283    1   51      0      0     324   154  C105
      415         1307       2   332    0   50      0      0     346     9   NaN
      416         1308       2   384    0   79      0      0     220    31   NaN
      417         1309       2   302    0   79      1      1     105    84   NaN
```

```
     Embarked
0           1
1           2
2           1
3           2
4           2
..        …
413         2
414         0
415         2
416         2
417         0

[418 rows x 11 columns]
```

[95]:

```python
df['SibSp'].value_counts()
```

[95]:

```
0    283
1    110
2     14
3      4
4      4
6      2
5      1
Name: SibSp, dtype: int64
```

[96]:

```python
y.value_counts()
```

[96]:

```
0    266
1    152
Name: Sex, dtype: int64
```

[97]:

```python
columns
```

[97]:

```
['Pclass', 'Name', 'Age', 'SibSp', 'Parch', 'Ticket', 'Fare', 'Embarked']
```

[101]:

```python
df
```

[101]:

```
     PassengerId  Pclass  Name  Age  SibSp  Parch  Ticket  Fare Cabin  \
0            892       2   206   44      0      0     152    24   NaN
1            893       2   403   60      1      0     221     5   NaN
2            894       1   269   74      0      0      73    41   NaN
3            895       2   408   34      0      0     147    34   NaN
4            896       2   178   27      1      1     138    46   NaN
..           …     …   …  …    …    …     …   …   …
413         1305       2   353   79      0      0     267    31   NaN
414         1306       0   283   51      0      0     324   154  C105
```

```
415         1307       2   332   50      0      0     346     9   NaN
416         1308       2   384   79      0      0     220    31   NaN
417         1309       2   302   79      1      1     105    84   NaN

     Embarked
0           1
1           2
2           1
3           2
4           2
..        …
413         2
414         0
415         2
416         2
417         0

[418 rows x 10 columns]
```

```
[102]: y
```

```
[102]: 0      0
       1      1
       2      0
       3      0
       4      1
             ..
       413    0
       414    1
       415    0
       416    0
       417    0
       Name: Sex, Length: 418, dtype: int64
```

```
[105]: x = df
```

```
[106]: x
```

```
[106]:      PassengerId  Pclass  Name  Age  SibSp  Parch  Ticket  Fare Cabin  \
       0            892       2   206   44      0      0     152    24   NaN
       1            893       2   403    60      1      0     221     5   NaN
       2            894       1   269   74      0      0      73    41   NaN
       3            895       2   408   34      0      0     147    34   NaN
       4            896       2   178   27      1      1     138    46   NaN
       ..           …      …    …   …    …    …     …    …   …
       413         1305       2   353   79      0      0     267    31   NaN
       414         1306       0   283   51      0      0     324   154  C105
```

```
415          1307       2   332   50      0      0     346     9   NaN
416          1308       2   384   79      0      0     220    31   NaN
417          1309       2   302   79      1      1     105    84   NaN

     Embarked
0           1
1           2
2           1
3           2
4           2
..        …
413         2
414         0
415         2
416         2
417         0

[418 rows x 10 columns]
```

[107]:

```
y
```

[107]:

```
0      0
1      1
2      0
3      0
4      1
      ..
413    0
414    1
415    0
416    0
417    0
Name: Sex, Length: 418, dtype: int64
```

[123]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

[125]:

```
x
```

[125]:

```
     PassengerId  Pclass  Name  Age  SibSp  Parch  Ticket  Fare  Embarked
0            892       2   206   44      0      0     152    24         1
1            893       2   403   60      1      0     221     5         2
2            894       1   269   74      0      0      73    41         1
3            895       2   408   34      0      0     147    34         2
4            896       2   178   27      1      1     138    46         2
..           …      …   …  …    …    …     …   …       …
413         1305       2   353   79      0      0     267    31         2
```

```
414         1306       0  283  51      0      0     324   154          0
415         1307       2  332  50      0      0     346     9          2
416         1308       2  384  79      0      0     220    31          2
417         1309       2  302  79      1      1     105    84          0

[418 rows x 9 columns]
```

```
[126]: y
```

```
[126]: 0      0
       1      1
       2      0
       3      0
       4      1
             ..
       413    0
       414    1
       415    0
       416    0
       417    0
       Name: Sex, Length: 418, dtype: int64
```

```
[127]: x_train , x_test , y_train , y_test = train_test_split(x ,y ,test_size=0.2 , random_state=42)
```

```
[128]: x_train
```

```
[128]:      PassengerId  Pclass  Name  Age  SibSp  Parch  Ticket  Fare  Embarked
       336         1228       1   413   40      0      0      79    50         2
       31           923       1   190   30      2      0     283   106         2
       84           976       1   221   79      0      0      72    43         1
       287         1179       0   351   30      1      0      52   149         2
       317         1209       1   319   24      0      0     122    42         2
       ..           …     …    …   …    …    …      …    …       …
       71           963       2   263   26      0      0     194    29         2
       106          998       2    52   26      0      0     153    23         1
       270         1162       0   252   59      0      0      33   143         0
       348         1240       1   157   30      0      0      82    52         2
       102          994       2   141   79      0      0     227    19         1

       [334 rows x 9 columns]
```

```
[129]: y_train
```

```
[129]: 336    0
       31     0
       84     0
```

```
287    0
317    0
      ..
71     0
106    0
270    0
348    0
102    0
Name: Sex, Length: 334, dtype: int64
```

[130]:

```
model = LogisticRegression()
```

[131]:

```
model.fit(x_train , y_train)
```

[131]:

```
LogisticRegression()
```

[132]:

```
y_pred = model.predict(x_test)
```

[133]:

```
from sklearn.metrics import accuracy_score , confusion_matrix ,␣
 ↪classification_report
```

[134]:

```
acc = accuracy_score(y_pred , y_test)
```

[135]:

```
print("Accuracy_score:", acc)
```

```
Accuracy_score: 0.5714285714285714
```

[136]:

```
conf_mat = confusion_matrix(y_pred , y_test)
```

[137]:

```
print("Confustion_matrix:", conf_mat)
```

```
Confustion_matrix: [[44 30]
 [ 6  4]]
```

[138]:

```
cla = classification_report(y_pred , y_test)
```

[139]:

```
print("Classification_repo:",cla)
```

```
Classification_repo:                precision    recall  f1-score   support

           0       0.88      0.59      0.71        74
           1       0.12      0.40      0.18        10

    accuracy                           0.57        84
   macro avg       0.50      0.50      0.45        84
weighted avg       0.79      0.57      0.65        84
```

[ ]: