* Coding: titanic crash survival
* GaussianNB classifier

**Berboulli: feaures are binary**

**Multinomial: features discrete, category**

**Gaussian: features continuous**

**import** pandas **as** pd

df **=** pd**.**read\_csv("titanic.csv")

df**.**head()

|  | **PassengerId** | **Name** | **Pclass** | **Sex** | **Age** | **SibSp** | **Parch** | **Ticket** | **Fare** | **Cabin** | **Embarked** | **Survived** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | Braund, Mr. Owen Harris | 3 | male | 22.0 | 1 | 0 | A/5 21171 | 7.2500 | NaN | S | 0 |
| **1** | 2 | Cumings, Mrs. John Bradley (Florence Briggs Th... | 1 | female | 38.0 | 1 | 0 | PC 17599 | 71.2833 | C85 | C | 1 |
| **2** | 3 | Heikkinen, Miss. Laina | 3 | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | NaN | S | 1 |
| **3** | 4 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | 1 | female | 35.0 | 1 | 0 | 113803 | 53.1000 | C123 | S | 1 |
| **4** | 5 | Allen, Mr. William Henry | 3 | male | 35.0 | 0 | 0 | 373450 | 8.0500 | NaN | S | 0 |

df**.**drop(['PassengerId','Name','SibSp','Parch','Ticket','Cabin','Embarked'],axis**=**'columns',inplace**=True**)

df**.**head()

|  | **Pclass** | **Sex** | **Age** | **Fare** | **Survived** |
| --- | --- | --- | --- | --- | --- |
| **0** | 3 | male | 22.0 | 7.2500 | 0 |
| **1** | 1 | female | 38.0 | 71.2833 | 1 |
| **2** | 3 | female | 26.0 | 7.9250 | 1 |
| **3** | 1 | female | 35.0 | 53.1000 | 1 |
| **4** | 3 | male | 35.0 | 8.0500 | 0 |

inputs **=** df**.**drop('Survived',axis**=**'columns')

target **=** df**.**Survived

*#inputs.Sex = inputs.Sex.map({'male': 1, 'female': 2})*

dummies **=** pd**.**get\_dummies(inputs**.**Sex)

dummies**.**head(3)

|  | **female** | **male** |
| --- | --- | --- |
| **0** | 0 | 1 |
| **1** | 1 | 0 |
| **2** | 1 | 0 |

inputs **=** pd**.**concat([inputs,dummies],axis**=**'columns')

inputs**.**head(3)

|  | **Pclass** | **Sex** | **Age** | **Fare** | **female** | **male** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | 3 | male | 22.0 | 7.2500 | 0 | 1 |
| **1** | 1 | female | 38.0 | 71.2833 | 1 | 0 |
| **2** | 3 | female | 26.0 | 7.9250 | 1 | 0 |

**I am dropping male column as well because of dummy variable trap theory. One column is enough to repressent male vs female**

inputs**.**drop(['Sex','male'],axis**=**'columns',inplace**=True**)

inputs**.**head(3)

|  | **Pclass** | **Age** | **Fare** | **female** |
| --- | --- | --- | --- | --- |
| **0** | 3 | 22.0 | 7.2500 | 0 |
| **1** | 1 | 38.0 | 71.2833 | 1 |
| **2** | 3 | 26.0 | 7.9250 | 1 |

inputs**.**columns[inputs**.**isna()**.**any()]

Index(['Age'], dtype='object')

inputs**.**Age[:10]

0 22.0

1 38.0

2 26.0

3 35.0

4 35.0

5 NaN

6 54.0

7 2.0

8 27.0

9 14.0

Name: Age, dtype: float64

inputs**.**Age **=** inputs**.**Age**.**fillna(inputs**.**Age**.**mean())

inputs**.**head()

|  | **Pclass** | **Age** | **Fare** | **female** |
| --- | --- | --- | --- | --- |
| **0** | 3 | 22.0 | 7.2500 | 0 |
| **1** | 1 | 38.0 | 71.2833 | 1 |
| **2** | 3 | 26.0 | 7.9250 | 1 |
| **3** | 1 | 35.0 | 53.1000 | 1 |
| **4** | 3 | 35.0 | 8.0500 | 0 |

**from** sklearn.model\_selection **import** train\_test\_split

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(inputs,target,test\_size**=**0.3)

**from** sklearn.naive\_bayes **import** GaussianNB

model **=** GaussianNB()

model**.**fit(X\_train,y\_train)

GaussianNB(priors=None, var\_smoothing=1e-09)

model**.**score(X\_test,y\_test)

Out[15]:

0.7835820895522388

X\_test[0:10]

Out[16]:

|  | **Pclass** | **Age** | **Fare** | **female** | **male** |
| --- | --- | --- | --- | --- | --- |
| **309** | 1 | 30.000000 | 56.9292 | 1 | 0 |
| **839** | 1 | 29.699118 | 29.7000 | 0 | 1 |
| **110** | 1 | 47.000000 | 52.0000 | 0 | 1 |
| **872** | 1 | 33.000000 | 5.0000 | 0 | 1 |
| **235** | 3 | 29.699118 | 7.5500 | 1 | 0 |
| **411** | 3 | 29.699118 | 6.8583 | 0 | 1 |
| **32** | 3 | 29.699118 | 7.7500 | 1 | 0 |
| **562** | 2 | 28.000000 | 13.5000 | 0 | 1 |
| **542** | 3 | 11.000000 | 31.2750 | 1 | 0 |
| **250** | 3 | 29.699118 | 7.2500 | 0 | 1 |

y\_test[0:10]

Out[17]:

309 1

839 1

110 0

872 0

235 0

411 0

32 1

562 0

542 0

250 0

Name: Survived, dtype: int64

model**.**predict(X\_test[0:10])

Out[18]:

array([1, 0, 0, 0, 1, 0, 1, 0, 1, 0], dtype=int64)

model**.**predict\_proba(X\_test[:10])

Out[19]:

array([[0.00455992, 0.99544008],

[0.91382024, 0.08617976],

[0.88164575, 0.11835425],

[0.92347978, 0.07652022],

[0.09084386, 0.90915614],

[0.99093305, 0.00906695],

[0.09094857, 0.90905143],

[0.97923786, 0.02076214],

[0.0516967 , 0.9483033 ],

[0.9909573 , 0.0090427 ]])

**Calculate the score using cross validation**

**from** sklearn.model\_selection **import** cross\_val\_score

cross\_val\_score(GaussianNB(),X\_train, y\_train, cv**=**5)

Out[34]:

array([0.75396825, 0.784 , 0.76612903, 0.82258065, 0.77419355])