GaussianNB

# Basics:



Figure 1. Imports

We will use variable **titanic** for storing our dataframe from **titanic.csv**, which we created in **prepare\_dataframe.py** (*preparing dataframe* in documentation):



Figure 2. Dataframe variable

We are trying to predict *‘survived’* column, so this will be our **target array**, or just **y**. Other columns form **feature matrix**, or **x**:



Figure 3. Target array



Figure 4. Feature matrix

For machine learning we will use **GaussianNB** model from **sklearn** module:



Figure 5. Model

For our model **estimating** we will use the following functions:



Figure 6. Estimating

# Learning and estimating:

1. Let’s start from estimating model using **train\_test\_split** and **accuracy\_score**:

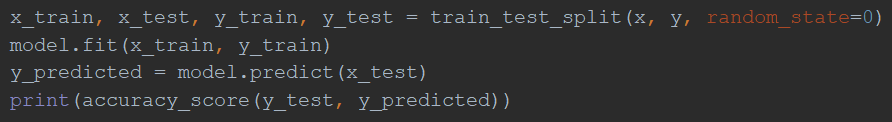


Figure 7. 'split' estimating

Accuracy\_score function compare **predicted** and **true** results. The output we get is **0.7982**. So, the accuracy of our model is nearly **80%**.

1. Now we use **cross\_val\_score**. Function got it’s name from evaluating method called **cross validation**.



Figure 8. Cross validation



Figure 9. cross\_val\_score output

If we will count the **mean** value of all those accuracies, we will get, again, nearly **80%**.

# Conclusion:

The result, our model **accuracy**, is pretty good for this model. **GaussianNB** is one of the simplest types of estimators. It doesn’t have a lot of parameters, so it is easy to set. I think that the result is great because we prepared our data well: we filled missing values and projected new features.