Logout 2_bayes Last Checkpoint: 6 hours ago (autosaved)

Menu

Python 2

- File
    - New Notebook
        - Python 2
        -
        - Python 3
    - Open...
    -
    - Make a Copy...
    - Rename...
    - Save and Checkpoint
    -
    - Revert to Checkpoint
        - Saturday, February 25, 2017 3:30 PM
    -
    - Print Preview
    - Download as
        - Notebook (.ipynb)
        - Python (.py)
        - HTML (.html)
        - Markdown (.md)
        - reST (.rst)
        - PDF via LaTeX (.pdf)
    -
    - Trusted Notebook
    -
    - Close and Halt
- Edit
    - Cut Cells
    - Copy Cells
    - Paste Cells Above

    - Paste Cells Below
    - Paste Cells & Replace
    - Delete Cells
    - Undo Delete Cells
    -
    - Split Cell
    - Merge Cell Above
    - Merge Cell Below
    -
    - Move Cell Up
    - Move Cell Down
    -
    - Edit Notebook Metadata
    -
    - Find and Replace
- View
    - Toggle Header
    - Toggle Toolbar
    - Cell Toolbar
        - None
        - Edit Metadata
        - Raw Cell Format
        - Slideshow
- Insert
    - Insert Cell Above
    - Insert Cell Below
- Cell
    - Run Cells
    - Run Cells and Select Below
    - Run Cells and Insert Below
    - Run All
    - Run All Above
    - Run All Below
    -
    - Cell Type
        - Code
        - Markdown

        - Raw NBConvert
    -
    - Current Outputs
        - Toggle
        - Toggle Scrolling
        - Clear
    - All Output
        - Toggle
        - Toggle Scrolling
        - Clear
- Kernel
    - Interrupt
    - Restart
    - Restart & Clear Output
    - Restart & Run All
    - Reconnect
    -
    - Change kernel
        - Python 2
        - Python 3
- Help
    - User Interface Tour
    - Keyboard Shortcuts
    -
    - Notebook Help
    - Markdown
    -
    - Python
    - IPython
    - NumPy
    - SciPy
    - Matplotlib
    - SymPy
    - pandas
    -
    - About

Code

CellToolbar

In [2]:

```
from sklearn import datasets

%pylab inline
Populating the interactive namespace from numpy and matplotlib
```

In [35]:

```
breast_cancer_data = datasets.load_breast_cancer()

digits_data = datasets.load_digits()
```

In [25]:

```
print ("features names: {}".format(breast_cancer_data['feature_names']))

print ("\nfeatures values for first element: {}".format(breast_cancer_data['data'][0]))

print ("\ntarget: {}".format(breast_cancer_data['target']))
features names: ['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
 'concave points error' 'symmetry error' 'fractal dimension error'
 'worst radius' 'worst texture' 'worst perimeter' 'worst area'
 'worst smoothness' 'worst compactness' 'worst concavity'
 'worst concave points' 'worst symmetry' 'worst fractal dimension']

features values for first element: [  1.79900000e+01   1.03800000e+01   1.22800000e+02   1.00100000e+03
   1.18400000e-01   2.77600000e-01   3.00100000e-01   1.47100000e-01
   2.41900000e-01   7.87100000e-02   1.09500000e+00   9.05300000e-01
   8.58900000e+00   1.53400000e+02   6.39900000e-03   4.90400000e-02
   5.37300000e-02   1.58700000e-02   3.00300000e-02   6.19300000e-03
   2.53800000e+01   1.73300000e+01   1.84600000e+02   2.01900000e+03
   1.62200000e-01   6.65600000e-01   7.11900000e-01   2.65400000e-01
   4.60100000e-01   1.18900000e-01]

target: [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 0 1 0 0 1 1 1 1 0 1 0 0
 1 0 1 0 0 1 1 1 0 0 1 0 0 0 1 1 1 0 1 1 0 0 1 1 1 0 0 1 1 1 1 0 1 1 0 1 1
 1 1 1 1 1 1 0 0 0 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 0 1 1 0 1 1 0 1 1 1 1 0 1
 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 1 1 0 0 0 1 0
 1 0 1 1 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 0 1 1 0 0 1 1
 1 0 1 1 1 1 1 0 0 1 1 0 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 0 1 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1
 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 0 0 0 1 1
 1 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0
 0 1 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1
 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0 1 1 1 1 1 0 1 1
 0 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1
 1 1 1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0 1 0 1 0 0
 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 1]
```

In [26]:

```
from sklearn.naive_bayes import BernoulliNB, MultinomialNB, GaussianNB

from sklearn.model_selection import cross_val_score
```

In [42]:

```
models = [BernoulliNB(), MultinomialNB(), GaussianNB()]

scores = []

for model in models:

    score = np.mean(cross_val_score(model, breast_cancer_data['data'], breast_cancer_data['target'], cv=5))

    scores.append(score)

    print "%s score for breast cancer dataset:  %f\n" % (str(model),score)

print "Best score for breast cancer dataset:  %f\n" % (np.max(score))
```

```
BernoulliNB(alpha=1.0, binarize=0.0, class_prior=None, fit_prior=True) score for breast cancer dataset:  0.627426
```

```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True) score for breast cancer dataset:  0.896376

GaussianNB(priors=None) score for breast cancer dataset:  0.940377

Best score for breast cancer dataset:  0.940377
```

In [38]:

```
print ("\nfeatures values for first element: {}".format(digits_data['images'][0]))

print ("\ntarget: {}".format(digits_data['target'][:40]))
```

```
features values for first element: [[  0.   0.   5.  13.   9.   1.   0.   0.]
 [  0.   0.  13.  15.  10.  15.   5.   0.]
 [  0.   3.  15.   2.   0.  11.   8.   0.]
 [  0.   4.  12.   0.   0.   8.   8.   0.]
 [  0.   5.   8.   0.   0.   9.   8.   0.]
 [  0.   4.  11.   0.   1.  12.   7.   0.]
 [  0.   2.  14.   5.  10.  12.   0.   0.]
 [  0.   0.   6.  13.  10.   0.   0.   0.]]

target: [0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 9 5 5 6 5 0
 9 8 9]
```

In [45]:

```
digits_stretched = [digits_data['images'][i].reshape(digits_data['images'][i].shape[0]*digits_data['images'][i].shape[1])

                    for i in range(len(digits_data['images']))]

scores = []

for model in models:

    score = np.mean(cross_val_score(model, digits_stretched, digits_data['target'], cv=5))

    scores.append(score)

    print "%s score for digits dataset:  %f\n" % (str(model),score)

print "Best score for digits dataset:  %f\n" % (np.max(scores))
```

```
BernoulliNB(alpha=1.0, binarize=0.0, class_prior=None, fit_prior=True) score for digits dataset:  0.824771

MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True) score for digits dataset:  0.871470
```

```
GaussianNB(priors=None) score for digits dataset:  0.806521

Best score for digits dataset:  0.871470



Best score for breast cancer dataset:  0.940377


(Было достигнуто с помощью наивного байесовского классификатора с нормальным распределением)


Best score for digits dataset:  0.871470


(Было достигнуто с помощью наивного байесовского классификатора с мультиномиальным распределением)




В итоге из исследования, проведённого для двух данных датасетов верны пункты: C, D
```

Best score for breast cancer dataset: 0.940377

(Было достигнуто с помощью наивного байесовского классификатора с нормальным распределением)

Best score for digits dataset: 0.871470

(Было достигнуто с помощью наивного байесовского классификатора с мультиномиальным распределением)

В итоге из исследования, проведённого для двух данных датасетов верны пункты: C, D