AtividadeMNIST-RandomForest

December 8, 2020

1 Lendo dados do MNIST

[1]:
```
from sklearn.datasets import fetch_openml
mnist = fetch_openml('mnist_784', version=1)
mnist.keys()
```

[1]:
```
dict_keys(['data', 'target', 'frame', 'categories', 'feature_names',
'target_names', 'DESCR', 'details', 'url'])
```

[2]:
```
28*28
```

[2]:
```
784
```

2 Conjunto de dados

[4]:
```
X, y = mnist["data"], mnist["target"]
X.shape, y.shape
```

[4]:
```
((70000, 784), (70000,))
```

3 Observando um caractere do MNIST

[5]:
```
import matplotlib as mpl
import matplotlib.pyplot as plt

some_digit = X[0]
some_digit_image = some_digit.reshape(28, 28)

plt.imshow(some_digit_image, cmap = mpl.cm.binary, interpolation="nearest")
plt.axis("off")
plt.show()
```

```
[6]: some_digit
```

```
[6]: array([  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   3.,  18.,
             18.,  18., 126., 136., 175.,  26., 166., 255., 247., 127.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
             30.,  36.,  94., 154., 170., 253., 253., 253., 253., 253., 225.,
            172., 253., 242., 195.,  64.,   0.,   0.,   0.,   0.,   0.,   0.,
              0.,   0.,   0.,   0.,   0.,  49., 238., 253., 253., 253., 253.,
            253., 253., 253., 253., 251.,  93.,  82.,  82.,  56.,  39.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
             18., 219., 253., 253., 253., 253., 253., 198., 182., 247., 241.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
              0.,   0.,   0.,   0.,   0.,   0.,   0.,  80., 156., 107., 253.,
            253., 205.,  11.,   0.,  43., 154.,   0.,   0.,   0.,   0.,   0.,
```

```
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,  14.,   1., 154., 253.,  90.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
139., 253., 190.,   2.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,  11., 190., 253.,  70.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,  35., 241., 225., 160., 108.,   1.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,  81., 240.,
253., 253., 119.,  25.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,  45., 186., 253., 253., 150.,  27.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,  16.,  93., 252., 253., 187.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0., 249., 253.,
249.,  64.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,  46., 130., 183., 253., 253., 207.,   2.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,  39., 148., 229., 253., 253., 253.,
250., 182.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,  24., 114.,
221., 253., 253., 253., 253., 201.,  78.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,  23.,  66., 213., 253., 253., 253., 253., 198.,  81.,
  2.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,  18., 171., 219., 253., 253.,
253., 253., 195.,  80.,   9.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,  55.,
172., 226., 253., 253., 253., 253., 244., 133.,  11.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0., 136., 253., 253., 253., 212., 135.,
132.,  16.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
  0.,   0.,   0.])
```

```
[7]: X[1].reshape(28, 28)
```

```
[7]: array([[  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
               0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
               0.,   0.,   0.,   0.,   0.,   0.],
            [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
               0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
               0.,   0.,   0.,   0.,   0.,   0.],
            [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
               0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
               0.,   0.,   0.,   0.,   0.,   0.],
            [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
               0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
               0.,   0.,   0.,   0.,   0.,   0.],
            [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
               0.,   0.,   0.,   0.,  51., 159., 253., 159.,  50.,   0.,   0.,
               0.,   0.,   0.,   0.,   0.,   0.],
            [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
               0.,   0.,   0.,  48., 238., 252., 252., 252., 237.,   0.,   0.,
               0.,   0.,   0.,   0.,   0.,   0.],
            [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
               0.,   0.,  54., 227., 253., 252., 239., 233., 252.,  57.,   6.,
               0.,   0.,   0.,   0.,   0.,   0.],
            [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
              10.,  60., 224., 252., 253., 252., 202.,  84., 252., 253., 122.,
               0.,   0.,   0.,   0.,   0.,   0.],
            [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
             163., 252., 252., 252., 253., 252., 252.,  96., 189., 253., 167.,
               0.,   0.,   0.,   0.,   0.,   0.],
            [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,  51.,
             238., 253., 253., 190., 114., 253., 228.,  47.,  79., 255., 168.,
               0.,   0.,   0.,   0.,   0.,   0.],
            [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,  48., 238.,
             252., 252., 179.,  12.,  75., 121.,  21.,   0.,   0., 253., 243.,
              50.,   0.,   0.,   0.,   0.,   0.],
            [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,  38., 165., 253.,
             233., 208.,  84.,   0.,   0.,   0.,   0.,   0.,   0., 253., 252.,
             165.,   0.,   0.,   0.,   0.,   0.],
            [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   7., 178., 252., 240.,
              71.,  19.,  28.,   0.,   0.,   0.,   0.,   0.,   0., 253., 252.,
             195.,   0.,   0.,   0.,   0.,   0.],
            [  0.,   0.,   0.,   0.,   0.,   0.,   0.,  57., 252., 252.,  63.,
               0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0., 253., 252.,
             195.,   0.,   0.,   0.,   0.,   0.],
            [  0.,   0.,   0.,   0.,   0.,   0.,   0., 198., 253., 190.,   0.,
               0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0., 255., 253.,
             196.,   0.,   0.,   0.,   0.,   0.],
```

```
       [  0.,   0.,   0.,   0.,   0.,   0.,  76., 246., 252., 112.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0., 253., 252.,
        148.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,  85., 252., 230.,  25.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.,   0.,   7., 135., 253., 186.,
         12.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,  85., 252., 223.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.,   7., 131., 252., 225.,  71.,
          0.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,  85., 252., 145.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,  48., 165., 252., 173.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,  86., 253., 225.,   0.,   0.,
          0.,   0.,   0.,   0., 114., 238., 253., 162.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,  85., 252., 249., 146.,  48.,
         29.,  85., 178., 225., 253., 223., 167.,  56.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,  85., 252., 252., 252., 229.,
        215., 252., 252., 252., 196., 130.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,  28., 199., 252., 252., 253.,
        252., 252., 233., 145.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,  25., 128., 252., 253.,
        252., 141.,  37.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.]])
```

4 Divida o dataset entre conjunto de treinamento e teste

Utilize 33% para o conjunto de teste

[43]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33)
```

5 Construa uma random forest e verifique a acurácia do modelo.

Utilize o algoritmos tradicional do `sklearn`

[44]:
```
from sklearn.ensemble import RandomForestClassifier
```

[45]:
```
randForest = RandomForestClassifier(n_estimators=100)
randForest.fit(X_train, y_train)
```

[45]:
```
RandomForestClassifier()
```

[46]:
```
y_pred = randForest.predict(X_test)
```

[47]:
```
from sklearn import metrics
```

[48]:
```
acc_rf = metrics.accuracy_score(y_test, y_pred)
print("Acurácia = ", acc_rf)
```

```
Acurácia =  0.9659740259740259
```

[49]:
```
acc_list = []
acc_list.append(acc_rf)
```

6 Construa uma random forest a partir dos métodos de bagging e pasting e compare a acurácia do modelo

[50]:
```
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
```

[51]:
```
bag_clas = BaggingClassifier(
    DecisionTreeClassifier(random_state=42), n_estimators=100,
    max_samples=100, bootstrap=True, n_jobs=-1, random_state=42)
bag_clas.fit(X_train, y_train)
```

[51]:
```
BaggingClassifier(base_estimator=DecisionTreeClassifier(random_state=42),
                  max_samples=100, n_estimators=100, n_jobs=-1,
                  random_state=42)
```

[52]:
```
y_pred = bag_clas.predict(X_test)
```

[53]:
```
acc_list.append(metrics.accuracy_score(y_test, y_pred))
```

[54]:
```
pasting_clas = BaggingClassifier(
    DecisionTreeClassifier(random_state=42), n_estimators=100,
    max_samples=100, bootstrap=False, n_jobs=-1, random_state=42)
pasting_clas.fit(X_train, y_train)
```

[54]:
```
BaggingClassifier(base_estimator=DecisionTreeClassifier(random_state=42),
                  bootstrap=False, max_samples=100, n_estimators=100, n_jobs=-1,
                  random_state=42)
```

[55]:
```
y_pred = pasting_clas.predict(X_test)
```

[56]:
```
acc_list.append(metrics.accuracy_score(y_test, y_pred))
```

[57]:
```
print("Random Forest:", acc_list[0])
print("Bagging:", acc_list[1])
print("Pasting:", acc_list[2])
```

```
Random Forest: 0.9659740259740259
Bagging: 0.8311688311688312
Pasting: 0.8301731601731601
```

7 Desafio

Altere os hyperparameters do modelo de random forest e verifique qual modelo apresenta melhor acurácia. Assim como se você tivesse que apresentar esses resultados para o ser cliente.

Teste os diferentes métodos de ensemble mostrados na disciplina.

- Bagging
- Pasting
- Boosting
- ...

[58]:
```
from sklearn.ensemble import AdaBoostClassifier

ada_clf = AdaBoostClassifier(
    DecisionTreeClassifier(max_depth=1), n_estimators=100,
    algorithm="SAMME.R", learning_rate=0.5, random_state=42)
ada_clf.fit(X_train, y_train)
```

[58]:
```
AdaBoostClassifier(base_estimator=DecisionTreeClassifier(max_depth=1),
                   learning_rate=0.5, n_estimators=100, random_state=42)
```

[59]:
```
y_pred = ada_clf.predict(X_test)
metrics.accuracy_score(y_test, y_pred)
```

[59]:
```
0.7967099567099567
```

[62]:
```
from sklearn.ensemble import GradientBoostingRegressor

gbrt = GradientBoostingRegressor(max_depth=2, n_estimators=3,
                                 learning_rate=1.0, random_state=42)
gbrt.fit(X_train, y_train)
```

[62]:
```
GradientBoostingRegressor(learning_rate=1.0, max_depth=2, n_estimators=3,
                          random_state=42)
```

[66]:
```
gbrt.score(X_test, y_test)
```

[66]:
```
0.4347195636119532
```