

0. sklearn

March 28, 2022

```
[1]: import numpy as np
import pandas as pd
from sklearn import datasets
from sklearn.model_selection import train_test_split

data = datasets.load_breast_cancer()
X = data.data
y = data.target
print("Dataset Shape: ",X.shape)
print("Frequency of labels:\n",pd.Series(y).value_counts())
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
```

Dataset Shape: (569, 30)

Frequency of labels:

1 357

0 212

dtype: int64

```
[2]: from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neural_network import MLPClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
```

```
[3]: knn = KNeighborsClassifier()
lr = LogisticRegression(solver='newton-cg')
mlp = MLPClassifier(max_iter=200)
svm = SVC()
nb = GaussianNB()
dtree = DecisionTreeClassifier()
rf = RandomForestClassifier()
ada = AdaBoostClassifier()
```

```
[4]: knn.fit(X_train,y_train)
y_pred = knn.predict(X_test)
accuracy = np.sum(y_test == y_pred)/len(y_test)
print("Training Accuracy: ", accuracy)
```

Training Accuracy: 0.9298245614035088

```
[5]: lr.fit(X_train,y_train)
y_pred = lr.predict(X_test)
accuracy = np.sum(y_test == y_pred)/len(y_test)
print("Training Accuracy: ", accuracy)
```

Training Accuracy: 0.9385964912280702

```
[6]: mlp.fit(X_train,y_train)
y_pred = mlp.predict(X_test)
accuracy = np.sum(y_test == y_pred)/len(y_test)
print("Training Accuracy: ", accuracy)
```

Training Accuracy: 0.9385964912280702

```
[7]: svm.fit(X_train,y_train)
y_pred=svm.predict(X_test)
accuracy = np.sum(y_test == y_pred)/len(y_test)
print("Training Accuracy: ", accuracy)
```

Training Accuracy: 0.9210526315789473

```
[8]: nb.fit(X_train,y_train)
y_pred = nb.predict(X_test)
accuracy = np.sum(y_test == y_pred)/len(y_test)
print("Training Accuracy: ", accuracy)
```

Training Accuracy: 0.9210526315789473

```
[9]: dtree.fit(X_train,y_train)
y_pred = dtree.predict(X_test)
accuracy = np.sum(y_test == y_pred)/len(y_test)
print("Training Accuracy: ", accuracy)
```

Training Accuracy: 0.9298245614035088

```
[10]: rf.fit(X_train,y_train)
y_pred = rf.predict(X_test)
accuracy = np.sum(y_test == y_pred)/len(y_test)
print("Training Accuracy: ", accuracy)
```

Training Accuracy: 0.9649122807017544

```
[11]: ada.fit(X_train,y_train)
      y_pred = ada.predict(X_test)
      accuracy = np.sum(y_test == y_pred)/len(y_test)
      print("Training Accuracy: ", accuracy)
```

Training Accuracy: 0.9649122807017544

```
[12]: #!pip install xgboost
      from xgboost import XGBClassifier
      xgb = XGBClassifier(use_label_encoder=False, objective = 'binary:logistic',
      ↪eval_metric = "logloss")
      xgb.fit(X_train, y_train)
      ypred = xgb.predict(X_test)
      y_pred = [round(value) for value in ypred]
      accuracy = np.sum(y_test == y_pred)/len(y_test)
      print("Training Accuracy: ", accuracy)
```

Training Accuracy: 0.9649122807017544

```
[ ]:
```