The classification challenge

Autumn 2020

```
################################################
##                                            ##
##  Machine Learning Fundamentals with Python  ##
##                                            ##
################################################
```

§1 Supervised Learning with scikit-learn

§1.1 Classification

§1.1.3 The classification challenge

1. What is the basic idea of the k-nearest neighbors algorithm (k-NN)?

- Predict the label of a data point by:
  - looking at the 'k' closet labeled data points;
  - taking a majority vote.

2. How to fit and predict by scikit-learn?

- Training a model on the data = 'fitting' a model to the data:
  - `.fit()` method
- To predict the labels of new data:
  - `.predict()` method

3. Code to fit a classifier by using scikit-learn:

[1]:
```
from sklearn import datasets
from sklearn.neighbors import KNeighborsClassifier

iris = datasets.load_iris()
knn = KNeighborsClassifier(n_neighbors=6)
knn.fit(iris['data'], iris['target'])
```

[1]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=6, p=2,
                     weights='uniform')
```

[2]:
```
iris['data'].shape
```

[2]: (150, 4)

[3]:
```
iris['target'].shape
```

[3]: (150,)

4. Code to predict on unlabeled data by using scikit-learn:

[4]:
```
import numpy as np

X_new = np.array([[5.6, 2.8, 3.9, 1.1], [5.7, 2.6, 3.8, 1.3],
                  [4.7, 3.2, 1.3, 0.2]])
prediction = knn.predict(X_new)
X_new.shape
```

[4]: (3, 4)

[5]:
```
print('Prediction: {}'.format(prediction))
```

```
Prediction: [1 1 0]
```

5. Practice exercise for the classification challenge:

▶ Data pre-loading:

[6]:
```
import pandas as pd

df = pd.read_csv(
    "https://archive.ics.uci.edu/ml/machine-learning-databases/voting-records/
 ↪house-votes-84.data",
    header=None)
df.columns = [
    'party', 'infants', 'water', 'budget', 'physician', 'salvador',
    'religious', 'satellite', 'aid', 'missile', 'immigration', 'synfuels',
    'education', 'superfund', 'crime', 'duty_free_exports', 'eaa_rsa'
]
df.replace(['y', 'n', '?'], [1, 0, 0.5], inplace=True)
X_new = pd.DataFrame([[
    0.69646919, 0.28613933, 0.22685145, 0.55131477, 0.71946897, 0.42310646,
    0.9807642, 0.68482974, 0.4809319, 0.39211752, 0.34317802, 0.72904971,
    0.43857224, 0.0596779, 0.39804426, 0.73799541
]])
```

▶ Fitting practice for k-nearest neighbors:

[7]:
```
# Import KNeighborsClassifier from sklearn.neighbors
from sklearn.neighbors import KNeighborsClassifier
```

```
# Create arrays for the features and the response variable
y = df['party'].values
X = df.drop('party', axis=1).values

# Create a k-NN classifier with 6 neighbors
knn = KNeighborsClassifier(n_neighbors=6)

# Fit the classifier to the data
knn.fit(X, y)
```

[7]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=6, p=2,
                     weights='uniform')
```

▶ **Predicting practice for k-nearest neighbors:**

[8]:
```
# Import KNeighborsClassifier from sklearn.neighbors
from sklearn.neighbors import KNeighborsClassifier

# Create arrays for the features and the response variable
y = df['party']
X = df.drop('party', axis=1)

# Create a k-NN classifier with 6 neighbors: knn
knn = KNeighborsClassifier(n_neighbors=6)

# Fit the classifier to the data
knn.fit(X, y)

# Predict the labels for the training data X
y_pred = knn.predict(X)

# Predict and print the label for the new data point X_new
new_prediction = knn.predict(X_new)
print("Prediction: {}".format(new_prediction))
```

```
Prediction: ['democrat']
```