Welcome to the course!

LINEAR CLASSIFIERS IN PYTHON

Michael (Mike) Gelbart

Instructor, The University of British Columbia

Assumed knowledge

In this course we'll assume you have some prior exposure to:

- Python, at the level of *Intermediate Python*
- scikit-learn, at the level of *Supervised Learning with scikit-learn*
- Supervised learning, at the level of *Supervised Learning with scikit-learn*

Fitting and predicting

```python
import sklearn.datasets

newsgroups = sklearn.datasets.fetch_20newsgroups_vectorized()

X, y = newsgroups.data, newsgroups.target
```

```python
X.shape
```

```
(11314, 130107)
```

```python
y.shape
```

```
(11314,)
```

Fitting and predicting (cont.)

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=1)
```

```
knn.fit(X,y)
```

```
y_pred = knn.predict(X)
```

Model evaluation

```
knn.score(X,y)
```

```
0.99991
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y)
```

```
knn.fit(X_train, y_train)

knn.score(X_test, y_test)
```

```
0.66242
```

Let's practice!

LINEAR CLASSIFIERS IN PYTHON

Applying logistic regression and SVM

LINEAR CLASSIFIERS IN PYTHON

Michael (Mike) Gelbart

Instructor, The University of British Columbia

Using LogisticRegression

```python
from sklearn.linear_model import LogisticRegression
```

```python
lr = LogisticRegression()
lr.fit(X_train, y_train)
lr.predict(X_test)
lr.score(X_test, y_test)
```

LogisticRegression example

```
import sklearn.datasets
wine = sklearn.datasets.load_wine()
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(wine.data, wine.target)
lr.score(wine.data, wine.target)
```

```
0.966
```

```
lr.predict_proba(wine.data[:1])
```

```
array([[9.966e-01, 2.740e-03, 6.787e-04]])
```

Using LinearSVC

`LinearSVC` works the same way:

```python
import sklearn.datasets

wine = sklearn.datasets.load_wine()
from sklearn.svm import LinearSVC

svm = LinearSVC()

svm.fit(wine.data, wine.target)
svm.score(wine.data, wine.target)
```

```
0.955
```

Using SVC

```python
import sklearn.datasets
wine = sklearn.datasets.load_wine()
from sklearn.svm import SVC
svm = SVC()
svm.fit(wine.data, wine.target);
svm.score(wine.data, wine.target)
```

```
0.708
```

Model complexity review:

- **Underfitting:** model is too simple, low training accuracy
- **Overfitting:** model is too complex, low test accuracy

Let's practice!

LINEAR CLASSIFIERS IN PYTHON

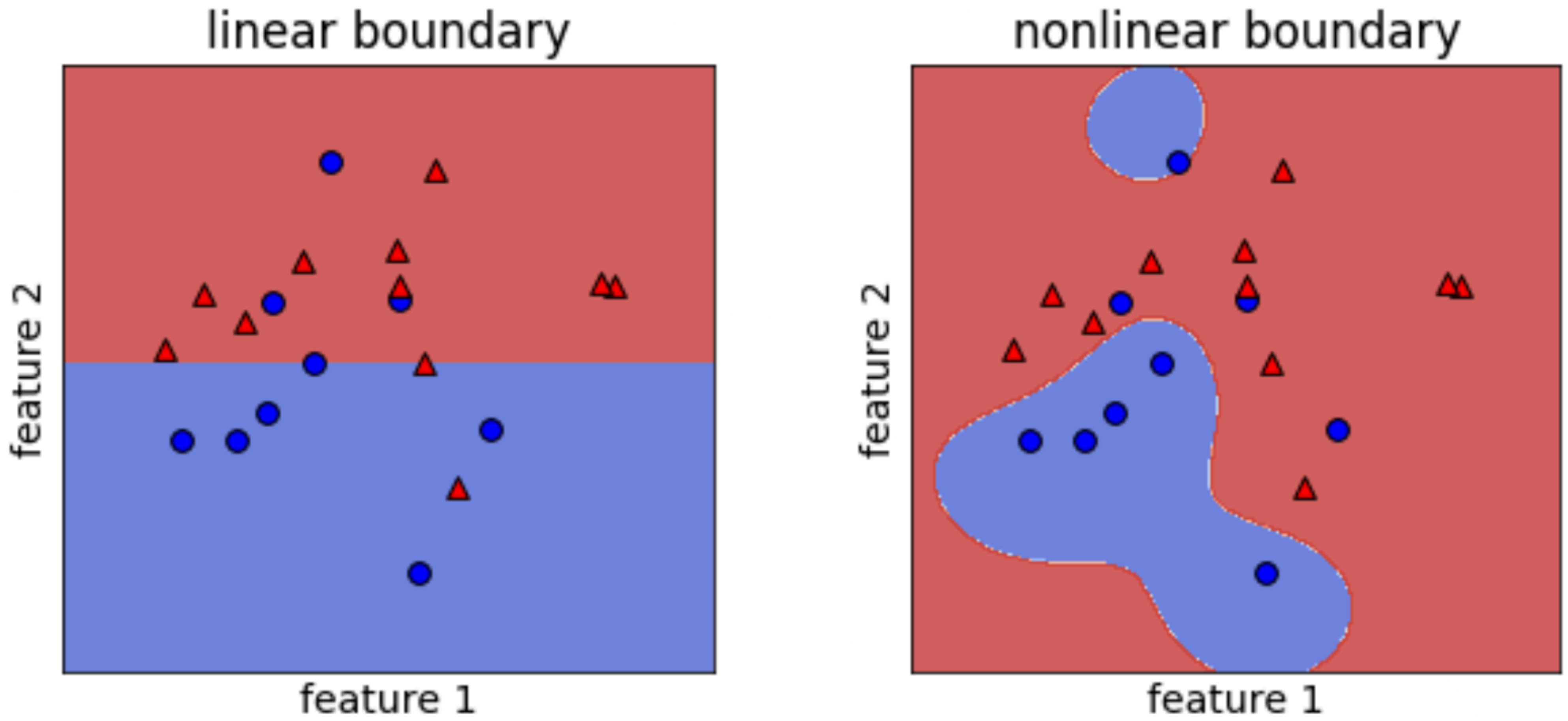
Linear decision boundaries

LINEAR CLASSIFIERS IN PYTHON

Michael (Mike) Gelbart

Instructor, The University of British Columbia

Linear decision boundaries

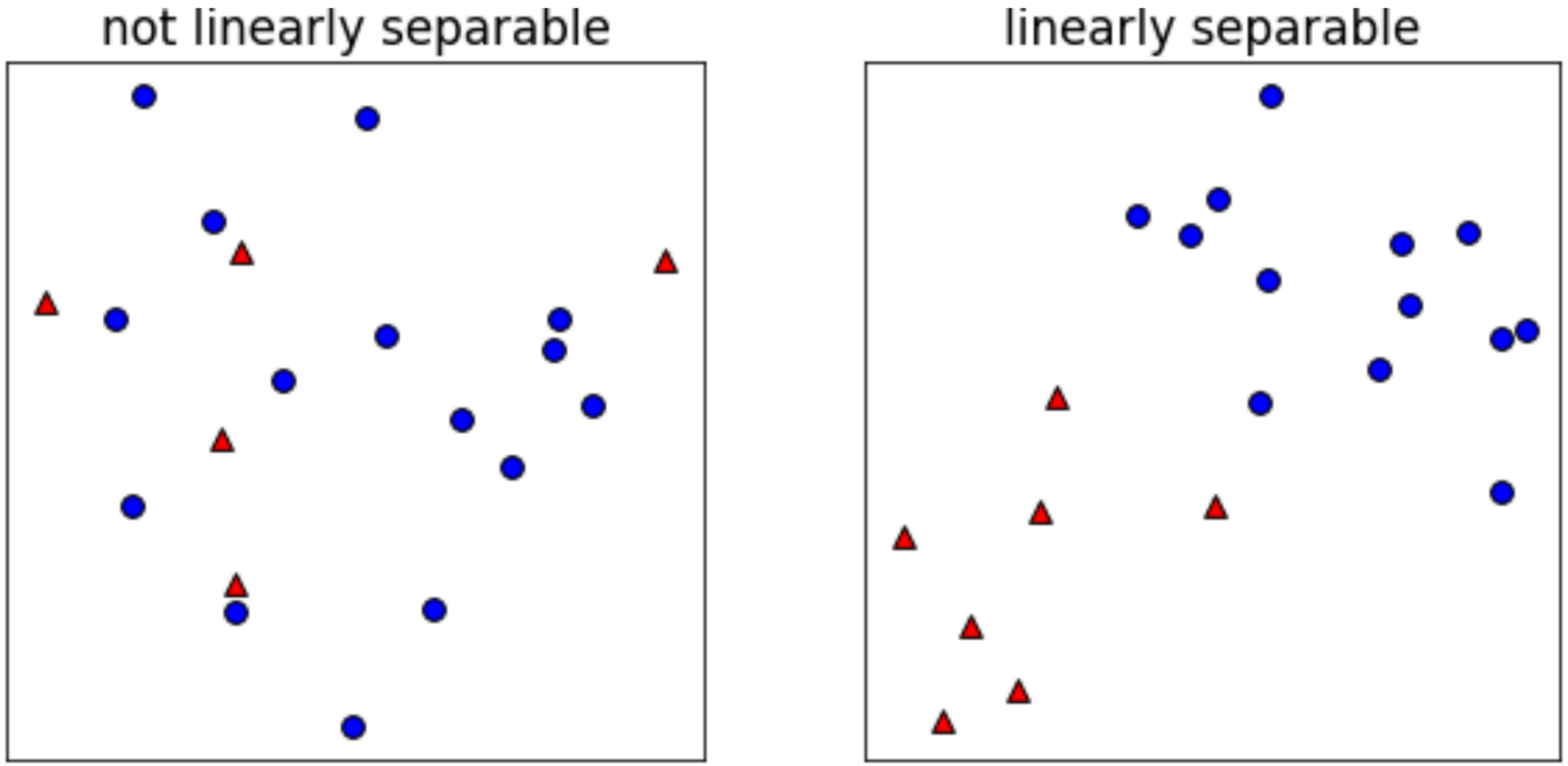


Definitions

Vocabulary:

- **classification:** learning to predict categories
- **decision boundary:** the surface separating different predicted classes
- **linear classifier:** a classifier that learns linear decision boundaries
  - e.g., logistic regression, linear SVM
- **linearly separable:** a data set can be perfectly explained by a linear classifier

Linearly separable data



Let's practice!

LINEAR CLASSIFIERS IN PYTHON