LINEAR CLASSIFIERS IN PYTHON

# Welcome to the course!

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 for Data Science*
- 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

```
In [1]: import sklearn.datasets

In [2]: newsgroups = sklearn.datasets.fetch_20newsgroups_vectorized()

In [3]: X, y = newsgroups.data, newsgroups.target
```

```
In [4]: X.shape
Out[4]: (11314, 130107)

In [5]: y.shape
Out[5]: (11314,)
```

```
In [6]: from sklearn.neighbors import KNeighborsClassifier

In [7]: knn = KNeighborsClassifier(n_neighbors=1)
```

```
In [8]: knn.fit(X,y)
```

```
In [9]: y_pred = knn.predict(X)
```

# Model evaluation

```
In [10]: knn.score(X,y)
Out[10]: 0.99991
```

```
In [11]: from sklearn.model_selection import train_test_split

In [12]: X_train, X_test, y_train, y_test = train_test_split(X, y)
```

```
In [13]: knn.fit(X_train, y_train)

In [14]: knn.score(X_test, y_test)
Out[14]: 0.66242
```

LINEAR CLASSIFIERS IN PYTHON

# Let's practice!

LINEAR CLASSIFIERS IN PYTHON

# Applying logistic regression and SVM

Michael (Mike) Gelbart
Instructor
The University of British Columbia

# Using LogisticRegression

```
In [1]: from sklearn.linear_model import LogisticRegression
```

```
In [2]: lr = LogisticRegression()

In [3]: lr.fit(X_train, y_train)

In [4]: lr.predict(X_test)

In [5]: lr.score(X_test, y_test)
```

# LogisticRegression example

```
In [1]: import sklearn.datasets

In [2]: wine = sklearn.datasets.load_wine()

In [3]: from sklearn.linear_model import LogisticRegression

In [4]: lr = LogisticRegression()

In [5]: lr.fit(wine.data, wine.target)

In [6]: lr.score(wine.data, wine.target)
Out[6]: 0.972

In [7]: lr.predict_proba(wine.data[:1])
Out[7]: array([[  9.951e-01,   4.357e-03,   5.339e-04]])
```

# Using LinearSVC

LinearSVC works the same way:

```
In [1]: import sklearn.datasets

In [2]: wine = sklearn.datasets.load_wine()

In [3]: from sklearn.svm import LinearSVC

In [4]: svm = LinearSVC()

In [5]: svm.fit(wine.data, wine.target)

In [6]: svm.score(wine.data, wine.target)
Out[6]: 0.893
```

# Using SVC

```
In [1]: import sklearn.datasets

In [2]: wine = sklearn.datasets.load_wine()

In [3]: from sklearn.svm import SVC

In [4]: svm = SVC() # default hyperparameters

In [5]: svm.fit(wine.data, wine.target);

In [6]: svm.score(wine.data, wine.target)
Out[6]: 1.
```

Model complexity review:

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

LINEAR CLASSIFIERS IN PYTHON

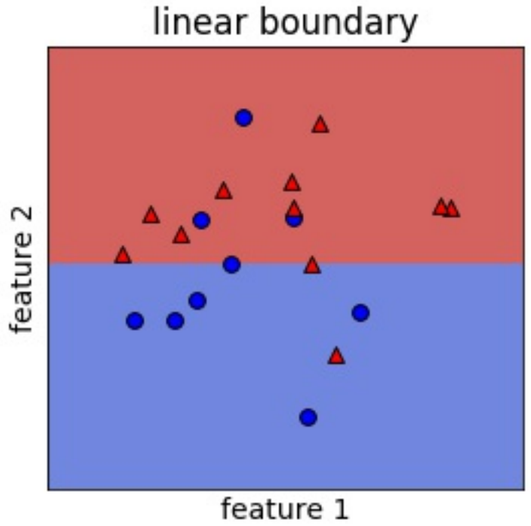
# Let's practice!

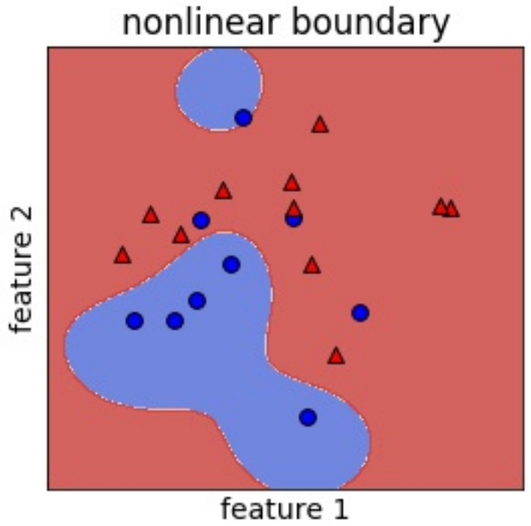
LINEAR CLASSIFIERS IN PYTHON

# Linear decision boundaries

Michael (Mike) Gelbart
Instructor
The University of British Columbia

# Linear decision boundaries



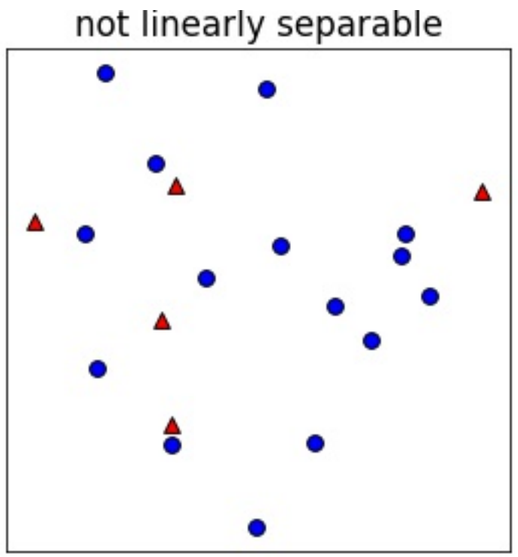



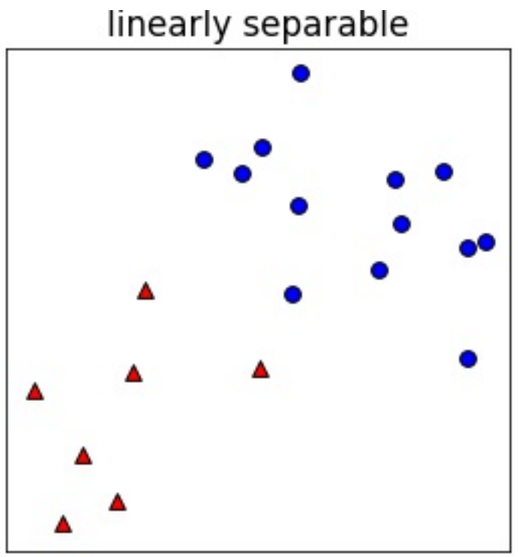
# Definitions

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



LINEAR CLASSIFIERS IN PYTHON

# Let's practice!