Classification

T.L. Grobler
(with some edits by S. Kroon)

Outline

- Introduction to supervised classification (labeled data):
  - Generative: Naive-Bayes
  - Discriminative: Logistic Regression

Data Sets & Preprocessing

- **Dimensionality reduction:**
  - PCA
  - LDA
- **Data sets – importance of data split:**
  - **Training set**: train classifier, preprocessing
  - **Validation set**: model selection, tune hyperparameters
  - **Test set**: evaluate final performance

Classification performance

- Single summary value: **Accuracy** (or loss)
- More nuanced view: **Confusion matrix**

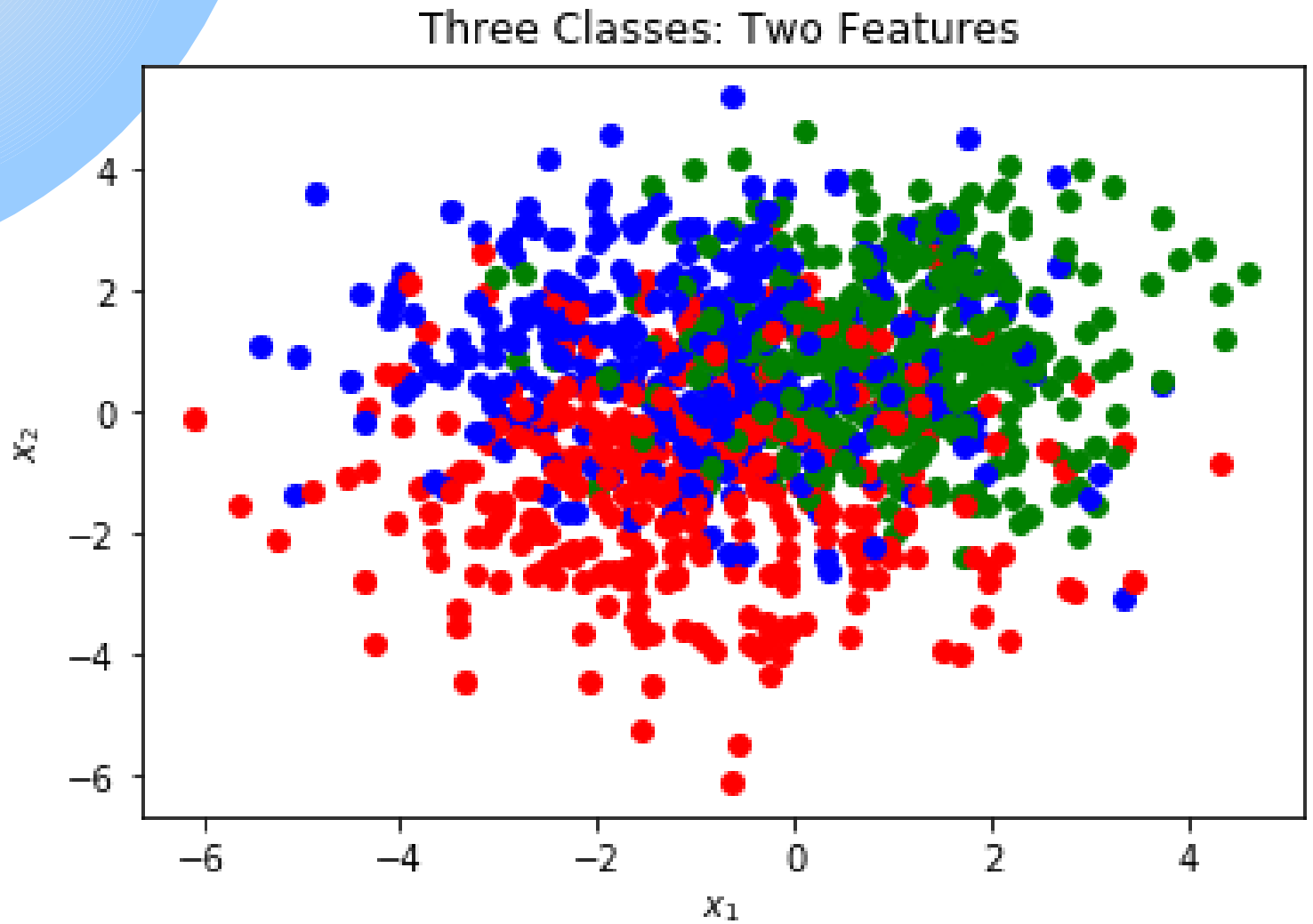
Classification

- Given **x** assign to one of k classes:
  - $C_j, j = 1,\ldots,k$
  - Assign prob $P(C_j|\mathbf{x})$
  - $C^* = \text{argmax}_{C_j} P(C_j|\mathbf{x})$
- Class probabilities are more useful than just knowing which class has the highest probability.

Data Description

- D: $(\mathbf{x}_j, y_j)$, $j = 1, \ldots, N$
  - observation $\mathbf{x}_j$ comes with class label y_j
  - $y_j = C_j$ if $\mathbf{x}_j \in C_j$
- Constructing $P(C_j|\mathbf{x})$ given D
- Two approaches: *generative* and *discriminative*

Example



$\mathbf{x} = [x_1, x_2]$ - features

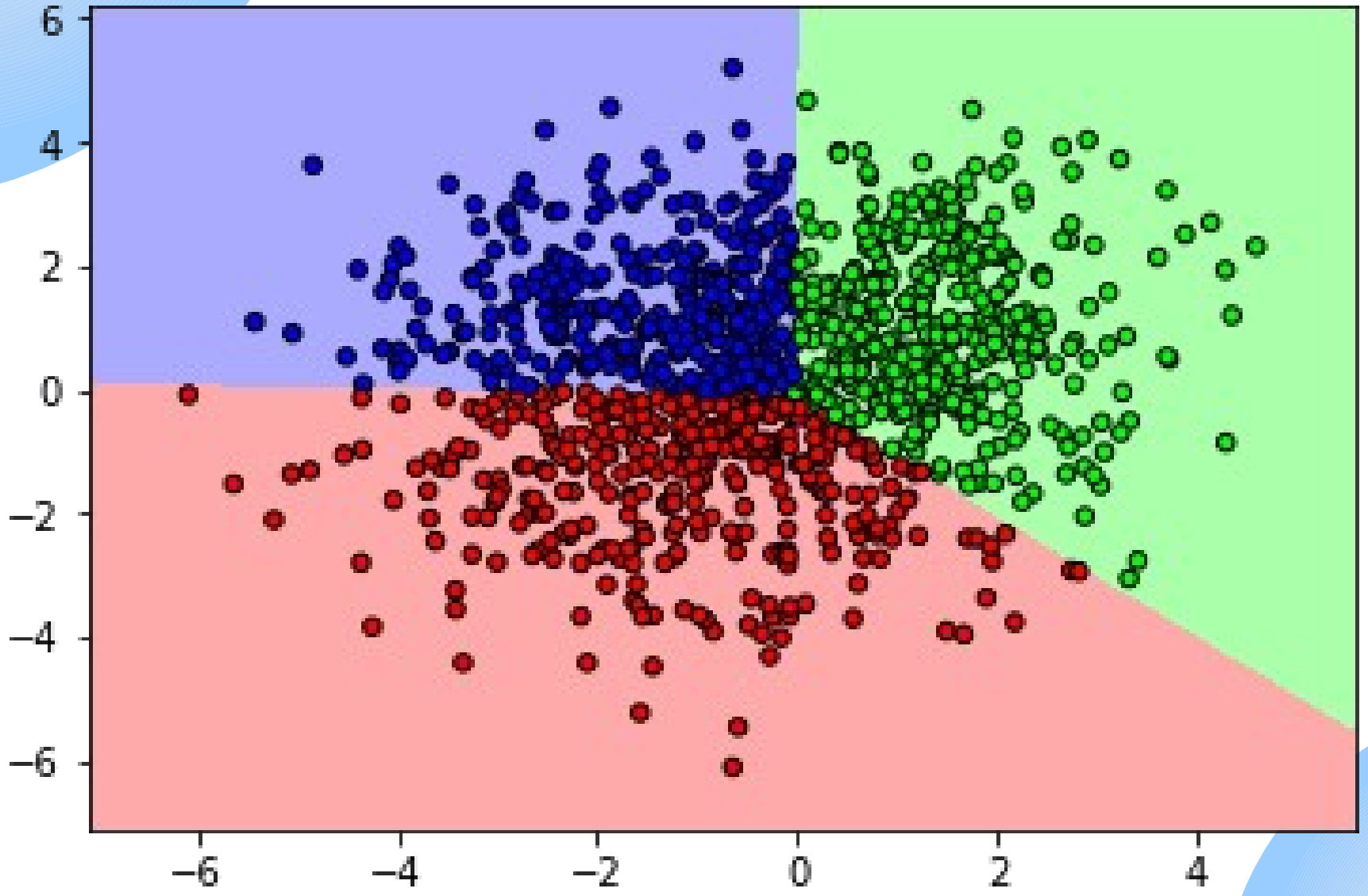
$C_1 = \text{r}$, $C_2 = \text{g}$ and $C_3 = \text{b}$

Autor: 02.03.22

Nearest Centroid Classifier

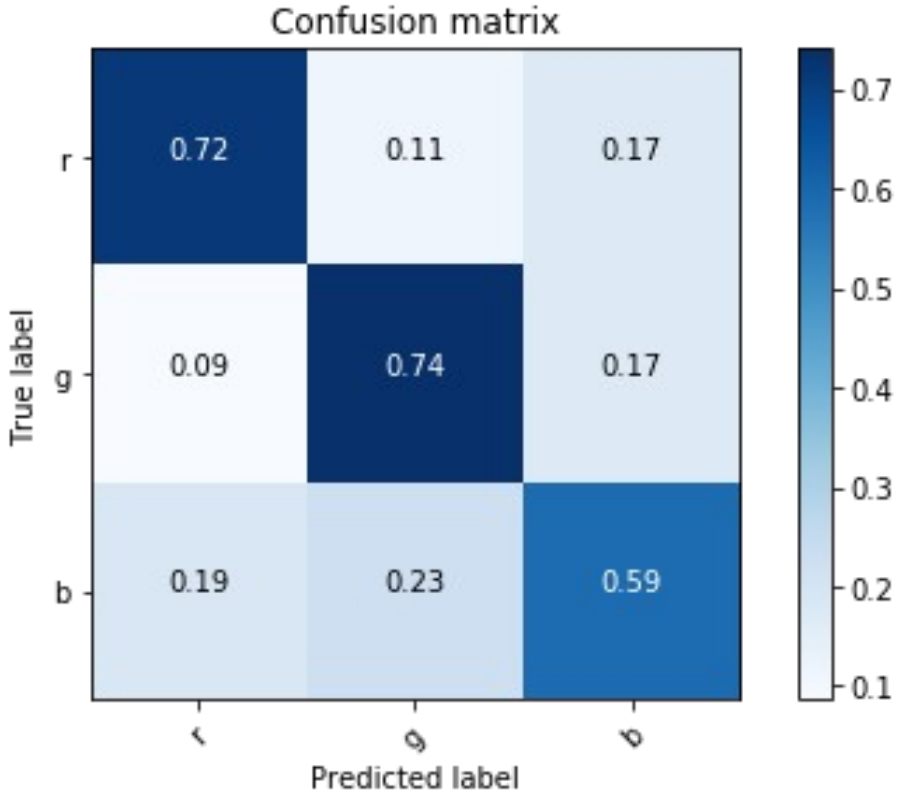
- Imports:
  - `from sklearn.neighbors import NearestCentroid`
  - `from sklearn.metrics import confusion_matrix`
- Train:
  - `clf = NearestCentroid()`
  - `clf.fit(X, y)`
- Predict:
  - `y_pred = clf.predict(X)`
- Confusion Matrix:
  - `cm = confusion_matrix(y,y_pred)`

Decision Boundary



Autor: 02.03.22

Confusion Matrix



M_{ij} is % of observations in class i predicted to be in group j.

Autor: 02.03.22

Generative Approach

- Estimate class-conditionals: $p(\mathbf{x}|C_j)$
- Posterior (Bayes Theorem):
  - $P(C_j|\mathbf{x}) \propto p(\mathbf{x}|C_j)P(C_j)$
  - Introduce $P(C_j)$
- Probabilistic Generative Models (PGM)

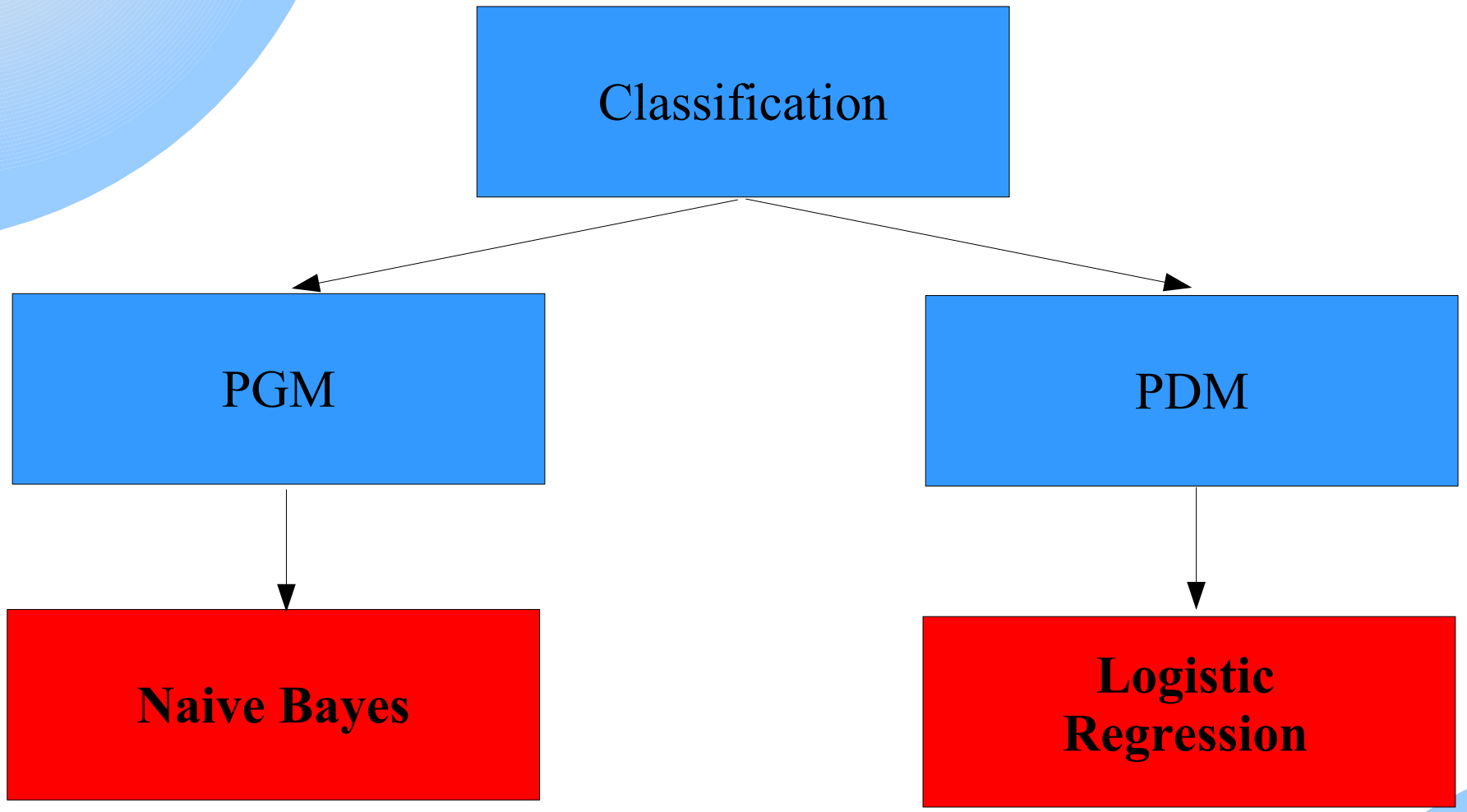
Discriminative Approach

- Dispenses with: $p(\mathbf{x}|C_j)$
- Directly compute posterior $P(C_j|\mathbf{x})$
- Probabilistic Discriminative Models (PDM)

Generative vs Discriminative

Generative	Discriminative
More Flexible	Less Flexible
Less Efficient for Classification	More Efficient for Classification
Training Simpler (per class)	Training Harder
Class Data	All Data
Models Each Class	Focuses on Class Differences

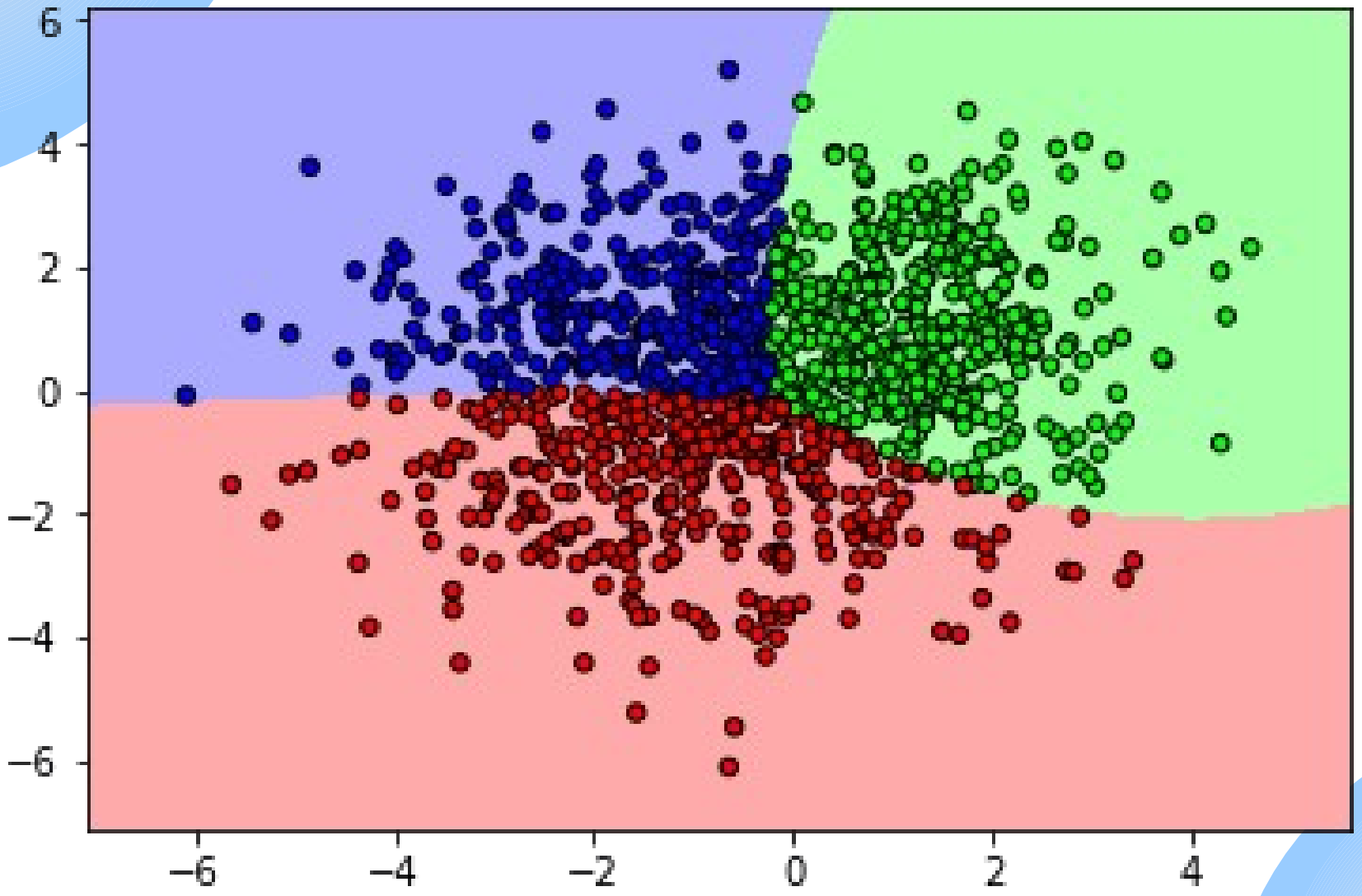
Specific models



Naive Bayes

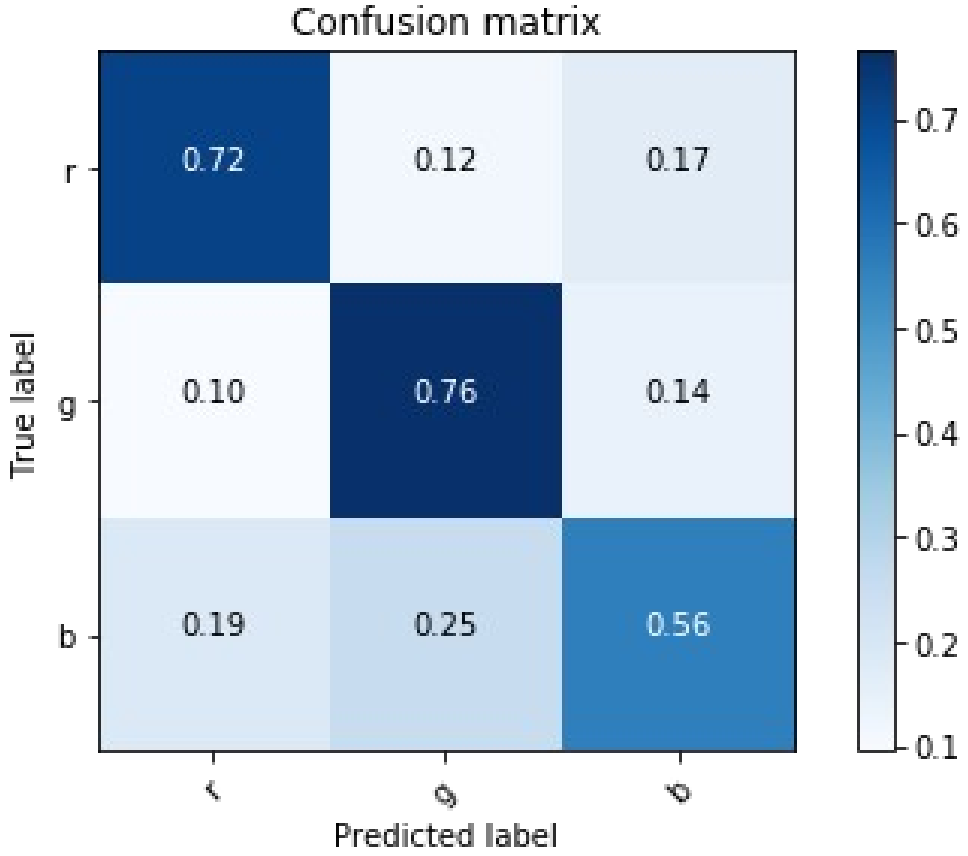
- Imports:
  - `from sklearn.naive_bayes import GaussianNB`
  - `from sklearn.metrics import confusion_matrix`
- Train:
  - `clf = GaussianNB()`
  - `clf.fit(X, y)`
- Predict:
  - `y_pred = clf.predict(X)`
- Confusion Matrix:
  - `cm = confusion_matrix(y,y_pred)`

Decision Boundary



Autor: 02.03.22

Confusion Matrix

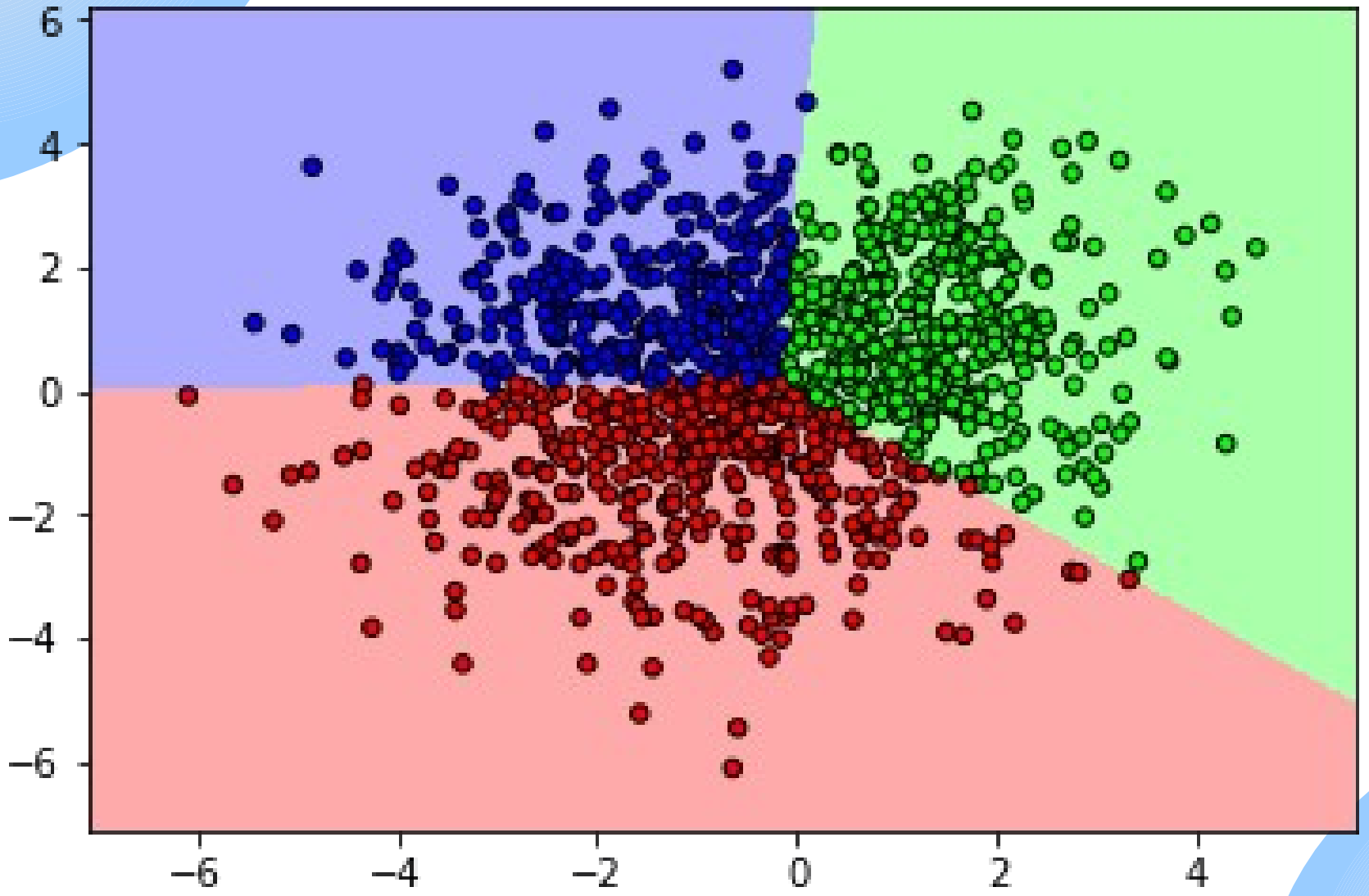


Autor: 02.03.22

Logistic Regression

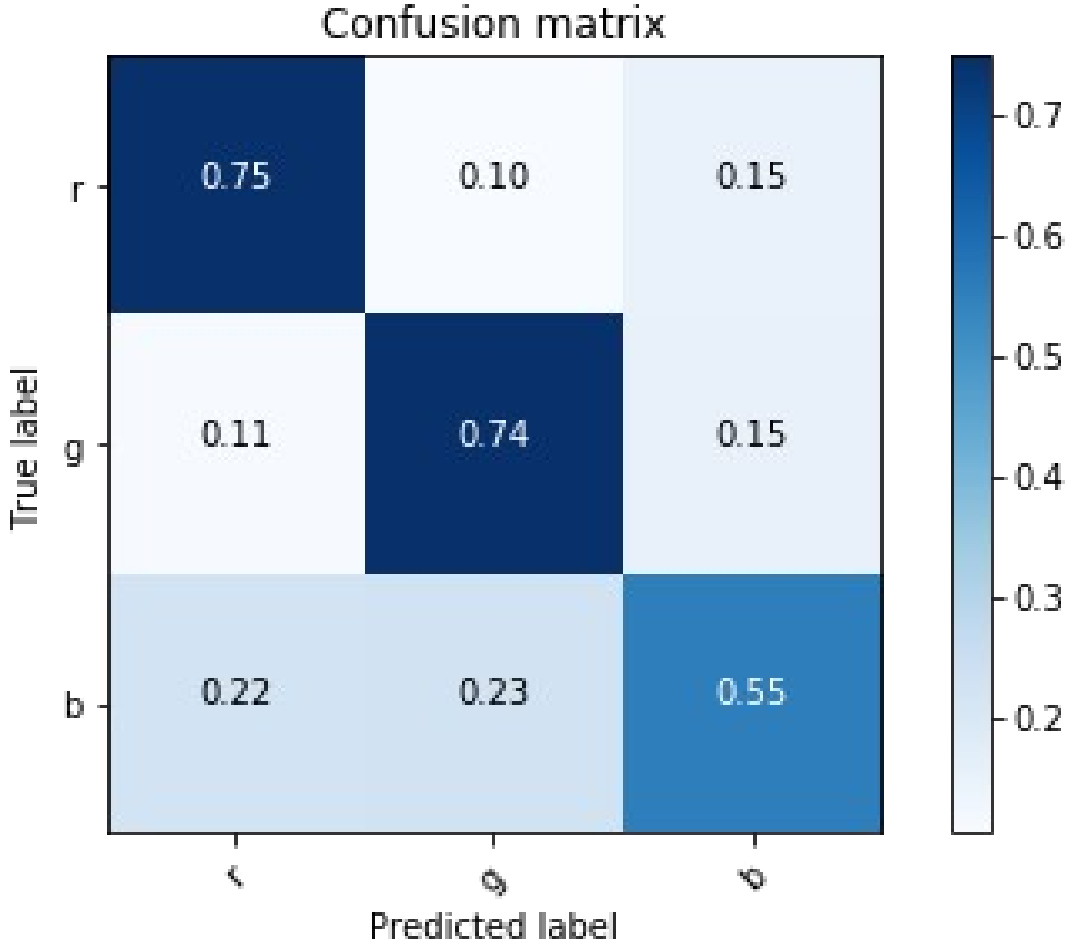
- Imports:
  - `from sklearn.linear_model import LogisticRegression as logis`
  - `from sklearn.metrics import confusion_matrix`
- Train:
  - `clf = linear_model.LogisticRegression(C=1e5)`
  - `clf.fit(X, y)`
- Predict:
  - `y_pred = clf.predict(X)`
- Confusion Matrix:
  - `cm = confusion_matrix(y,y_pred)`

Decision Boundary



Autor: 02.03.22

Confusion Matrix



Autor: 02.03.22