

# FUNDAMENTOS DE APRENDIZAJE AUTOMÁTICO

## PRÁCTICA 4: CONJUNTOS DE CLASIFICADORES

Irene Rodríguez Luján

# PRÁCTICA 4

- **Objetivo:** Diseño e implementación de **conjuntos de clasificadores** por **voto por mayoría** y comparación con la librería scikit-learn.
  - Clasificadores a utilizar: Naive Bayes (P1), Vecinos Próximos (P2) y Regresión Logística (P2).
- **Tareas:**
  1. Implementar la clase **ClasificadorEnsemble**
    - Métodos entrenamiento y clasifica
  2. Implementar el conjunto de clasificadores de clasificadores utilizando la clase **VotingClassifier** o **EnsembleVoteClassifier** de Scikit-learn.

# CONJUNTOS DE DATOS

- **Digits**
- Tres conjuntos de clasificadores a elegir por el alumno:
  - Repositorio **UCI Machine Learning**: <https://archive.ics.uci.edu/ml/datasets.html>
  - Otros repositorios → consultar al profesor
  - Puede ser necesario **preprocesar** los datos para que el fichero de entrada tenga el formato especificado
  - Recordar que Vecinos Próximos y Regresión Logística solo se implementaron para **atributos continuos** y que para clasificar problemas de más de dos clases con Regresión Logística es necesario utilizar el *wrapper* **ClasificadorMulticlase**.

# CLASIFICADORENSEMBLE

- Implementación del conjunto de clasificadores con **voto por mayoría**.
- Pueden utilizarse los clasificadores propios (ClasificadorNaiveBayes, ClasificadorVecinosPropios, ClasificadorRegresionLogistica) o las implementaciones proporcionadas por Scikit-learn (GaussianNB, KNeighborsClassifier, LogisticRegression).

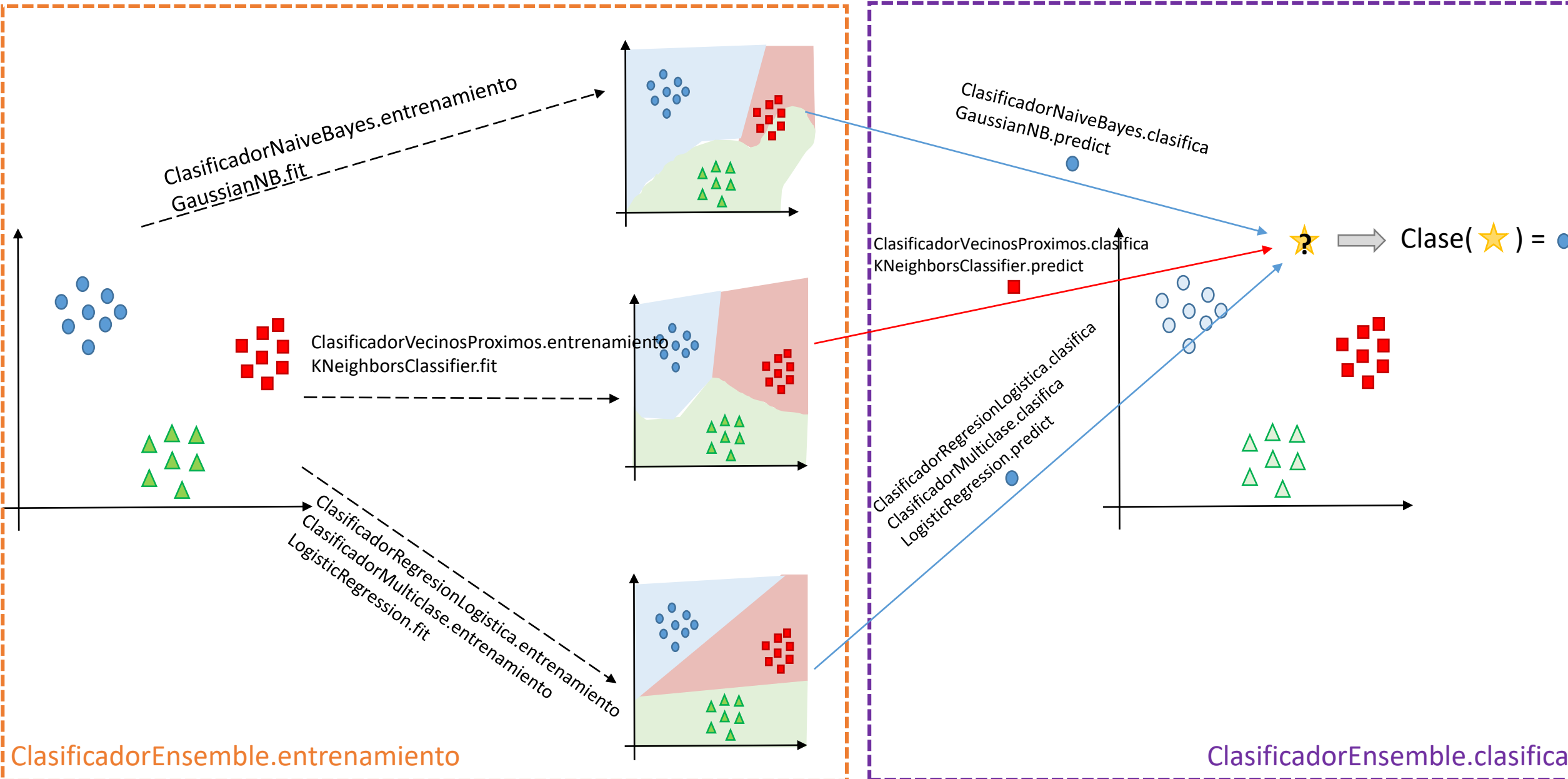
# CLASIFICADORENSEMBLE

- **Instanciación:** puede ser útil fijarse en la interfaz de VotingClassifier de Scikit-learn.

```
model = VotingClassifier(estimators=[('lr', clf1), ('gnb', clf2)], voting='hard')
```

- **Entrenamiento:** entrenar cada uno de los algoritmos del ensemble de acuerdo a ciertos parámetros de configuración.
- **Clasifica:** la predicción de la clase para cada uno de los patrones de test se determina como la clase más votada entre cada uno de los clasificadores que forman en conjunto.
- Para la implementación puede ser útil echar un vistazo a las funciones de entrenamiento y clasificación de ClasificadorMulticlase.

# CLASIFICADORENSEMBLE



# SCIKIT-LEARN

## VOTINGCLASSIFIER / ENSEMBLEVOTECLASSIFIER

- Ejemplo de uso:

Scikit-learn >  
0.17

```
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import VotingClassifier
```

```
X = np.array([[ -1, -1], [-2, -1], [-3, -2], [1, 1],
               [2, 1], [3, 2]])
y = np.array([1, 1, 1, 2, 2, 2])
```

```
clf1 = LogisticRegression(random_state=1)
clf2 = GaussianNB()
```

```
model = VotingClassifier(estimators=[
    ('lr', clf1), ('gnb', clf2)], voting='hard')
```

```
...
model.fit(X_train, y_train)
predicted= model.predict(X_test)
```

Github  
(autor de  
VotingClassifier)

```
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from mlxtend.classifier import EnsembleVoteClassifier
```

```
X = np.array([[ -1, -1], [-2, -1], [-3, -2], [1, 1],
               [2, 1], [3, 2]])
y = np.array([1, 1, 1, 2, 2, 2])
```

```
clf1 = LogisticRegression(random_state=1)
clf2 = GaussianNB()
```

```
model = EnsembleVoteClassifier(clfs=[clf1, clf2, clf3],
                               voting='hard')
```

```
...
model.fit(X_train, y_train)
predicted= model.predict(X_test)
```

# ENTREGA DE PRÁCTICA

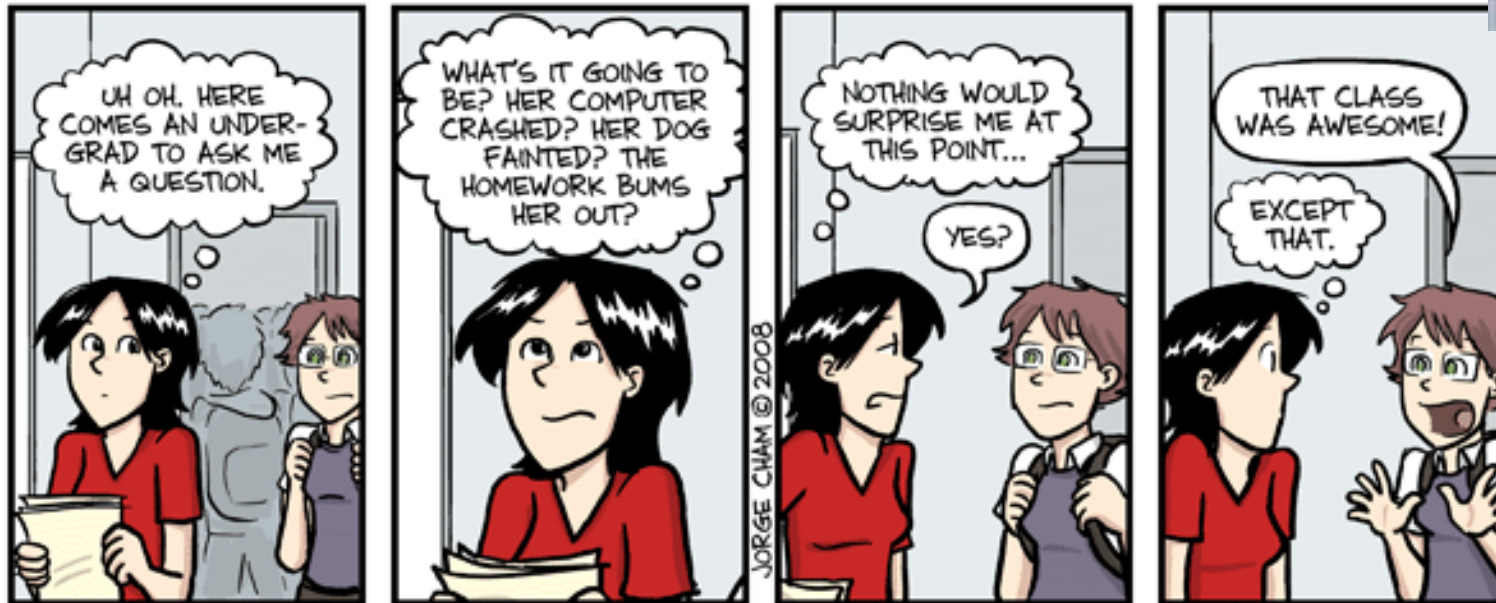
- Domingo 18 de diciembre a las 23.59h.
- Notebook (.ipnb, .html) + \*.py
- **OJO! Normalización de datos** (¿sobre qué datos se calcula la media y la desviación? ¿qué pasa si la desviación típica es 0?)
- **¡IMPORTANTE!** Incluir en el Notebook las tablas con los resultados obtenidos para las diferentes pruebas...

Dataset	MultinomialNB	MultinomialNB (Laplace)	GaussianNB
tic-tac-toe	29.68 ± 3.26	29.71 ± 3.27	32.5 ± 2.95
wine	15.69 ± 7.72	15.97 ± 7.55	2.63 ± 2.05
crx_clean	46.87 ± 6.42	32.29 ± 3.97	15.95 ± 2.58
digits	22.89 ± 1.98	22.89 ± 1.98	48.67 ± 3.03

- ... y **analizar** los resultados!



# NECESITAMOS PATRONES DE ENTRENAMIENTO...



NECESITAMOS PATRONES  
DE ENTRENAMIENTO...

