Raúl Garreta
Tryolabs / Fing Udelar
@raulgarreta

# Aprendizaje Automático con Python

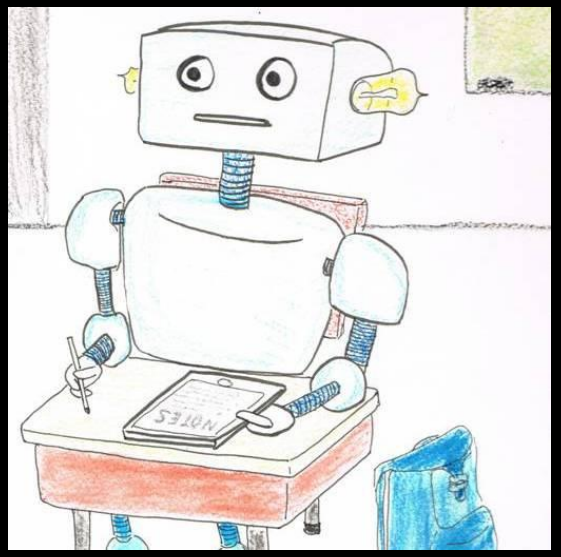
PyCon Uruguay 2012

# Agenda

- ¿Qué es Aprendizaje Automático (AA) ?
- ¿Qué se puede hacer con AA?
- Herramientas de AA en Python
- Ejemplos

# Aprendizaje Automático

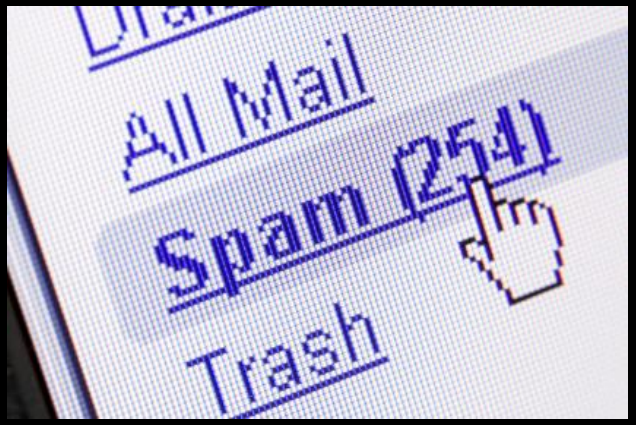
- Subárea dentro de Inteligencia Artificial.
- Estudia algoritmos que tienen la capacidad de aprender a realizar una tarea automáticamente.
- Mejoran su performance con la experiencia.
- Permiten resolver tareas complejas, cuya solución es muy difícil o imposible de realizar manualmente.
- Aprendizaje como aspecto fundamental en la Inteligencia.



# Aplicaciones

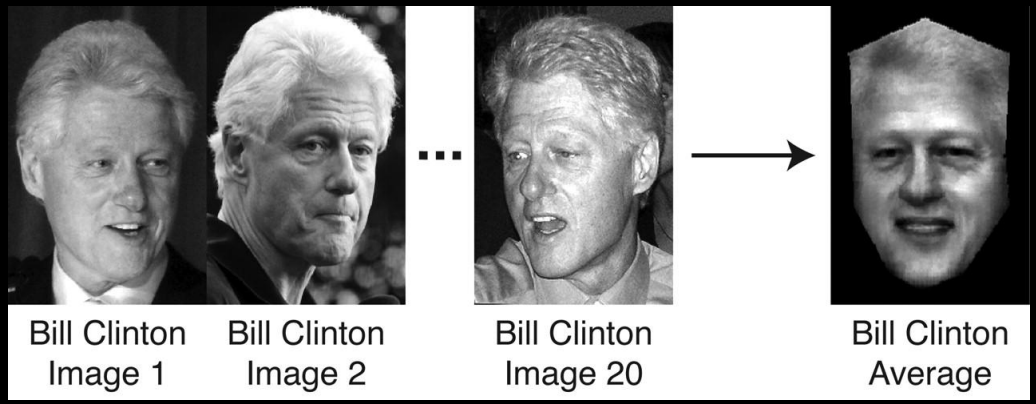
- Procesamiento de Lenguaje Natural


- Spam Filtering

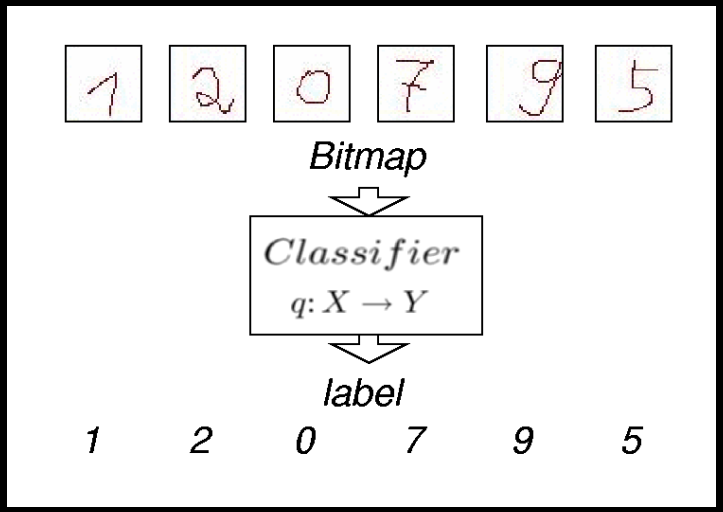


# Aplicaciones

- Visión Artificial
  - Reconocimiento de Rostros



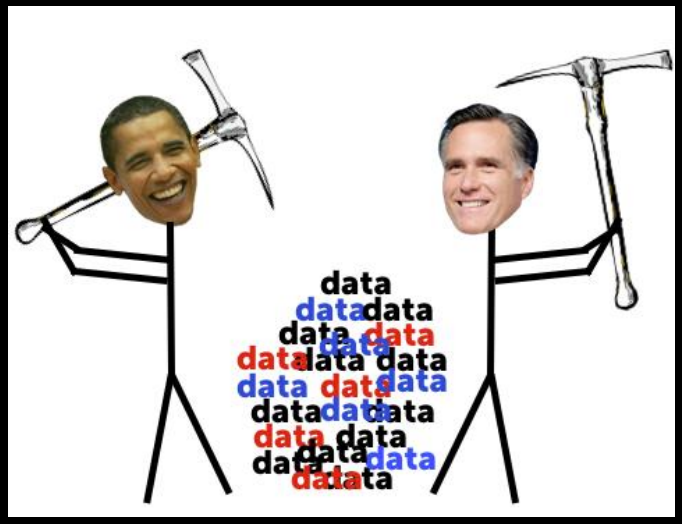

  - OCR



# Aplicaciones

- Jugadores Artificiales

- Data Mining



# Aplicaciones

- Sistemas de Recomendación

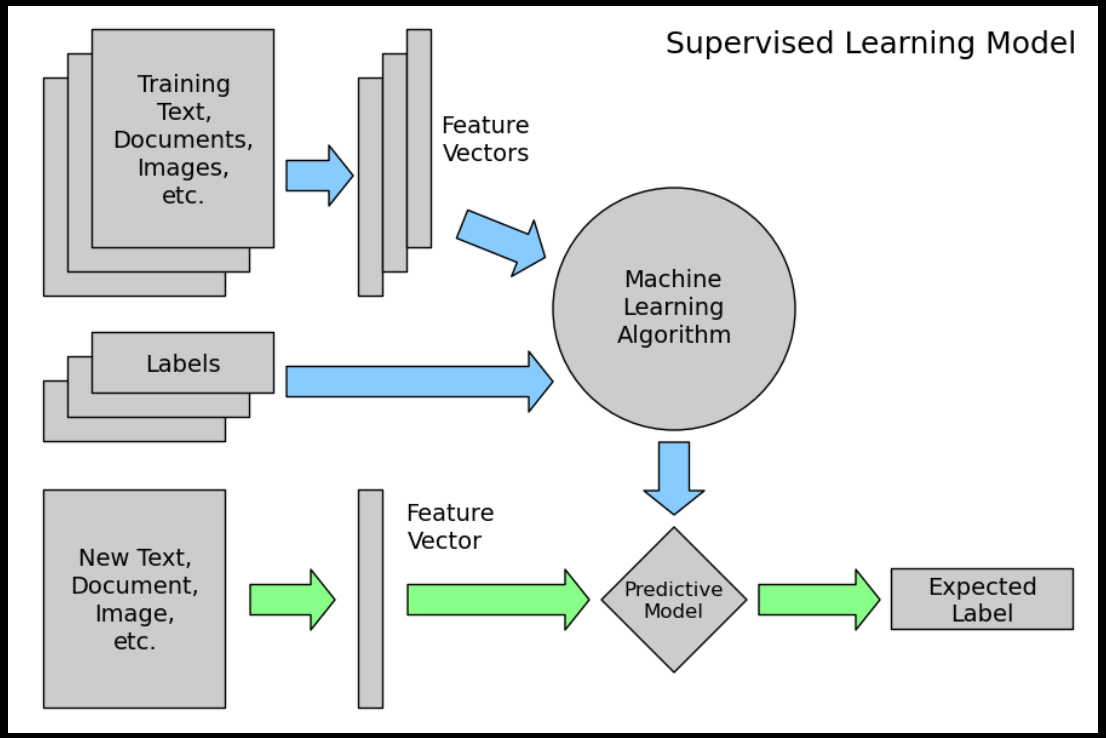

- Y mucho, mucho más…

# Definición Formal

- Mejorar en una tarea **T**
- Respecto a una medida de performance **P**
- Basándose en la experiencia **E**

# Tipo de Aprendizaje / Algoritmos

- Supervisado
  - Clasificación
    - Árboles de decisión
    - Naive Bayes
    - SVM
    - ...
  - Regresión
    - Redes Neuronales
    - ...
- No Supervisado
  - Clustering
    - KNN
    - SOM
- Por Refuerzos
  - Temporal Difference



# Ejemplo: Aprender a filtrar Spam

- T: clasificar mails en **Spam / No Spam**
- P: porcentaje de mails correctamente clasificados
- E: ver una muestra de mails clasificados manualmente por el usuario como **Spam / No Spam**

# Ejemplo

- ¿Qué es lo que se aprende y cómo se modela?
  - V: Mail-> {Spam, No Spam}
  - V(*m1*) = Spam, si *m1* es mail de spam
  - V(*m2*) = No Spam, si *m2* es mail de interés
  - V = *f*(aparece la palabra "viagra", el remitente está en mi lista de contactos?, #que aparece la palabra "compre", …)
  - *f* = función lineal? Función polinomial de 2do grado? Red neuronal? Árbol de Decisión? …

# Ejemplo

- ¿Con qué algoritmo se aprende?
  - Esto muchas veces depende de la representación/modelo que se va a utilizar
- Si modelo con una **Red Neuronal** -> puedo utilizar **Backpropagation**
- Si modelo con un **Árbol de Decisión** -> puedo utilizar **ID3**
- Si utilizo un modelo probabilístico -> estimar las probabilidades contando frecuencias.
- …

# Ejemplo

- Qué tipo de entrenamiento se utiliza?
- Supervisado: tengo ejemplos etiquetados, una base de mails ya clasificados como spam / no spam.
- Utilizo esta base como conjunto de entrenamiento.
- Puedo particionar en entrenamiento / testeo para aprender y testear respectivamente. Ejemplo: 70% para entrenamiento, 30% testeo

# Otros aspectos importantes

- Feature Selection: cual es el input del algoritmo, como represento un ejemplo, cuales son las características importantes a considerar para clasificar.
- Medidas de Performance: ¿cómo sé si el sistema realmente funciona bien? ¿cómo sé si el sistema mejora si realizo modificaciones?
  - Train set / Testing set
  - Precision, Recall, Medida F
  - Matriz de Confusión

# Herramientas en Python

- Hay muchas opciones:
  - Orange http://orange.biolab.si/
  - NLTK http://nltk.org/
  - Mlpy http://mlpy.sourceforge.net/
  - Pyml http://pyml.sourceforge.net/
  - Pybrain http://pybrain.org/
  - Scikit-learn http://scikit-learn.org/

# Ejemplo en Python

- ¿Cómo implementar nuestro spam filter en Python en 6 pasos sencillos?
- Utilizaremos Scikit-learn

# Paso 1: Conseguir Ejemplos

- Necesitamos recolectar ejemplos de entrenamiento.
- Mails etiquetados como spam / ham
- Exportar mis mails de mi cuenta de gmail

messages
    ham
        ham1.txt
        ham2.txt
        …
    spam
        spam1.txt
        spam2.txt
        …

# Paso 2: Cargar Ejemplos en Memoria

```
data_samples =
  load_files(container_path='/path/to/messages',
  shuffle=True)
```

# Paso 3: Particionar Entrenamiento/Testeo

```
SPLIT_PERC = 0.6
train_size = int(len(data_samples.data)*SPLIT_PERC)
data_train = data_samples.data[:train_size]
data_test = data_samples.data[train_size:]

y_train = data_samples.target[:train_size]
y_test = data_samples.target[train_size:]
```

# Paso 4: Preprocesar Ejemplos

```
vectorizer = TfidfVectorizer(sublinear_tf=True,
  strip_accents='ascii')


x_train = vectorizer.fit_transform(data_train)

x_test = vectorizer.transform(data_test)
```

# Paso 5: Entrenar el Clasificador

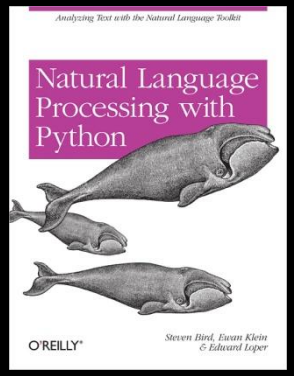
```
classifier = MultinomialNB()

classifier.fit(x_train, y_train)
```

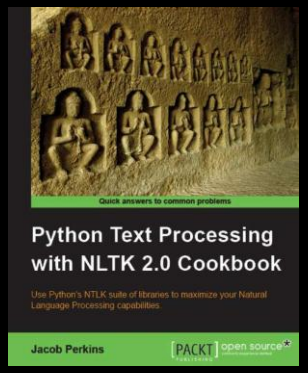
# Paso 6: Probar el Clasificador

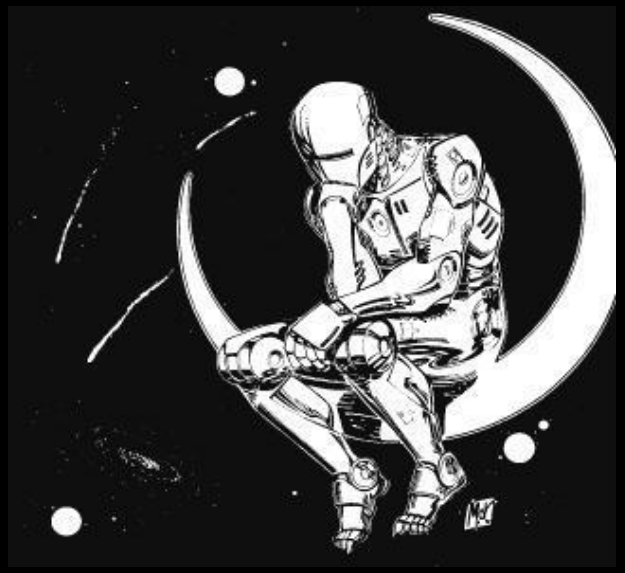
```
pred = classifier.predict(x_test)


metrics.precision_score(y_test, pred)

metrics.recall_score(y_test, pred)

metrics.f1_score(y_test, pred)

metrics.confusion_matrix(y_test, pred)
```

# Recursos

- Web:
  - Scikit-learn
    http://scikit-learn.org/stable/auto_examples
  - Streamhacker.com
- Libros:
  - NLTK book, NLTK cookbook
  - Machine Learning, Tom Mitchel
- Cursos:
  - Udelar
    http://www.fing.edu.uy/inco/cursos/aprendaut/
  - Stanford University
    https://www.coursera.org/course/ml
  - Washington University
    https://www.coursera.org/course/machlearning



# ¿Preguntas?