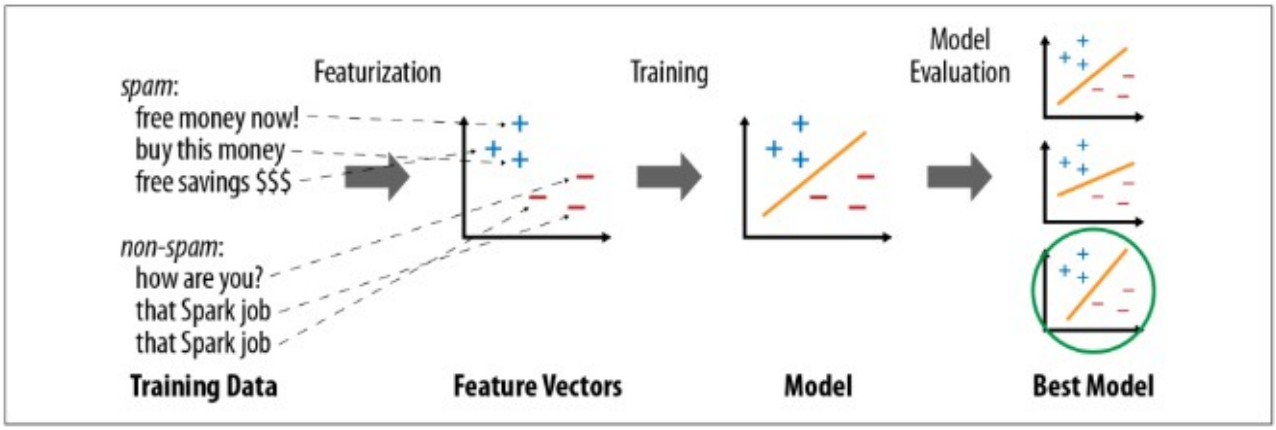
Ejemplo clasificación Spam con mLlib



El objetivo de los algoritmos de machine learning es tomar decisiones basadas en datos de entrenamiento.

Por ejemplo,clasificación de spam, donde a partir de un conjunto de datos de entrada(dataset) el objetivo es determinar si un email se puede clasificar como spam

Como entrada tenemos 2 ficheros,uno con ejemplos de emails que son spam y otro con ejemplos con no lo son

Para resolver este problema podemos usar los algoritmos **HashingTF** y **LogisticRegressionWithSGD** .

- **HashingTF** extrae las características a partir de los ficheros de entrada
- **LogisticRegressionWithSGD** es el algoritmo encargado de obtener el mejor modelo a partir de los datos de entrenamiento

```
from pyspark import SparkContext
from pyspark.mllib.regression import LabeledPoint
from pyspark.mllib.classification import LogisticRegressionWithSGD
from pyspark.mllib.feature import HashingTF
```

El código fuente de la clase LogisticRegressionWithSGD se puede ver en el repositorio:

https://apache.googlesource.com/spark/+/master/python/pyspark/mllib/classification.py

El código fuente de la clase HashingTF se puede ver en el repositorio:

https://apache.googlesource.com/spark/+/master/python/pyspark/mllib/feature.py

Se llama al método transform de la clase HaashingTF que convierte el documento de entrada en vectores de frecuencia de términos.

```
@since('1.2.0')
def transform(self, document):
    """
    Transforms the input document (list of terms) to term frequency
    vectors, or transform the RDD of document to RDD of term
    frequency vectors.
    """
    if isinstance(document, RDD):
        return document.map(self.transform)

    freq = {}
    for term in document:
        i = self.indexOf(term)
        freq[i] = 1.0 if self.binary else freq.get(i, 0) + 1.0
    return Vectors.sparse(self.numFeatures, freq.items())
```