

Machine Learning- Featurization

Prajwol Sangat

Updated by CM Ting (15 April 2022)



Machine Learning: Pipeline

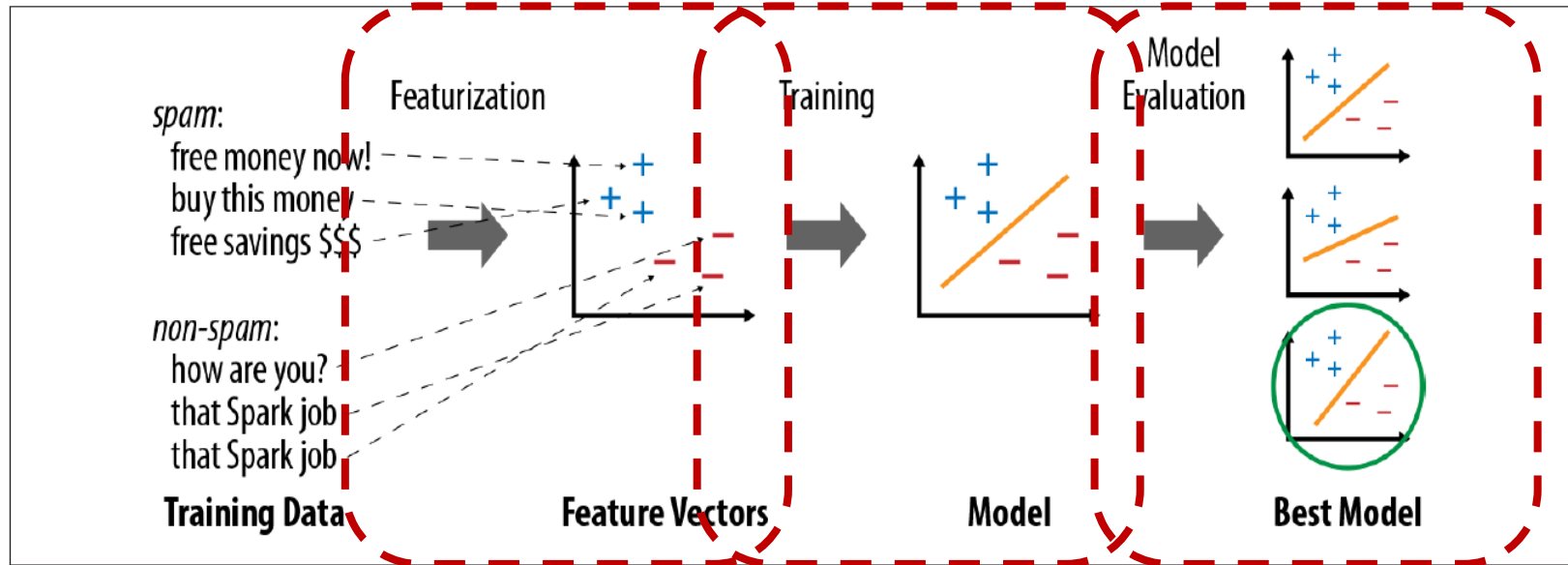


Figure 11-1. Typical steps in a machine learning pipeline

Machine Learning: Pipeline

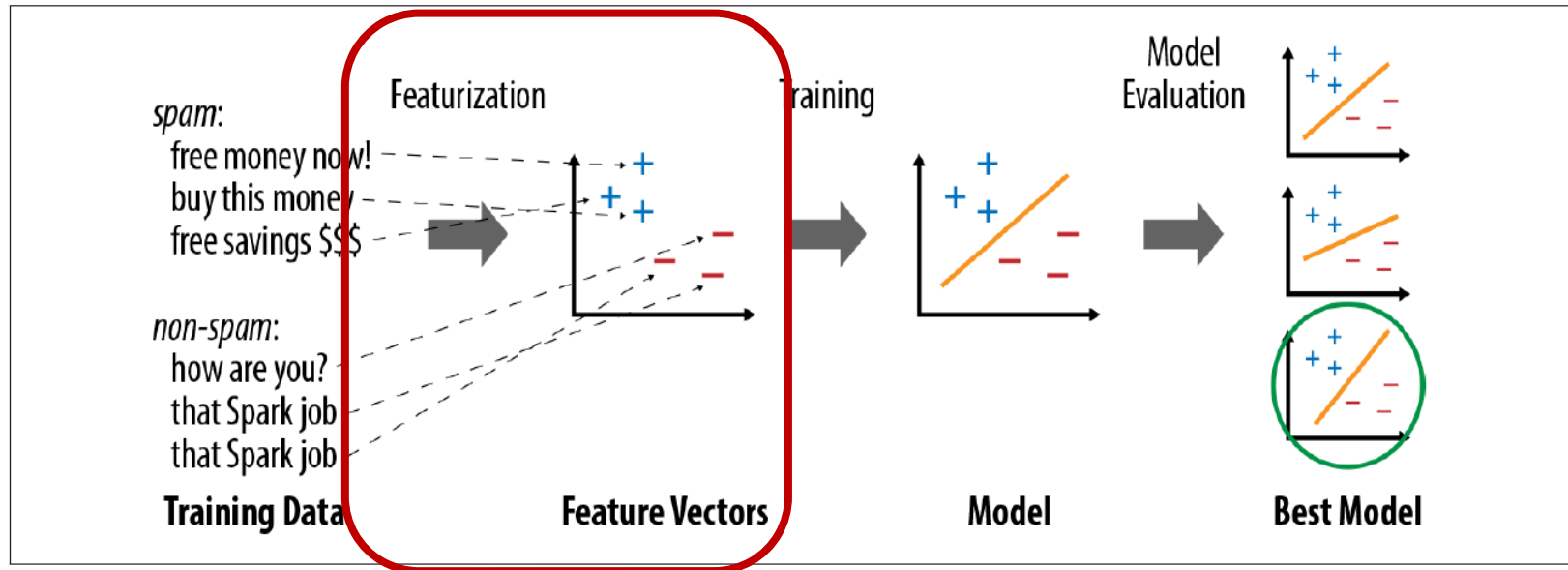


Figure 11-1. Typical steps in a machine learning pipeline

Featurization: **Extraction, transformation and selection**

- **Extraction**

- Extracting features from “raw” data

- **Transformation**

- Scaling, converting, or modifying features

- **Selection**

- Selecting a subset from a larger set of features

Featurization: Feature Extraction and Transformation

■ Features

- Any machine learning algorithm requires some training data. In training data we have values for all features for all historical records. Consider this simple data set

Height	Weight	Age	Class
165	70	22	Male
160	58	22	Female

- We can prepare training data by following two techniques
 - *Feature Extraction*
 - *Feature Selection*

Featurization: **Feature Extraction and Transformation**

■ **Feature extractors**

- CountVectorizer
- TF-IDF
- Word2Vec
- FeatureHasher (In tutorial)

Featurization: **Feature Extractors**

■ **Count Vectorizer**

- Convert a collection of text documents to vectors of token counts.
- During the fitting process, *Count Vectorizer* will select the top *vocabSize* words ordered by term frequency across the corpus.

```
id | texts
---|-----
0  | Array("a", "b", "c")
1  | Array("a", "b", "b", "c", "a")
```

Featurization: **Feature Extractors**

- **Term Frequency–Inverse Document Frequency, or TF-IDF,**
 - A simple way to generate feature vectors from text documents (e.g., web pages).
 - It computes two statistics for each term in each document:
 - *The term frequency (TF)*, which is the number of times the term occurs in that document, and
 - *The inverse document frequency (IDF)*, which measures how (in)frequently a term occurs across the whole document corpus.

Featurization: **Feature Extractors**

- **Term Frequency–Inverse Document Frequency, or TF-IDF,**
 - Denote a term by ***t***, a document by ***d***, and the corpus by ***D***.
 - Term frequency ***TF(t, d)*** is the number of times that term ***t*** appears in document ***d***, while document frequency ***DF(t, D)*** is the number of documents that contains term ***t***.
 - Inverse document frequency is a numerical measure of how much information a term provides:

$$IDF(t, D) = \log \frac{|D| + 1}{DF(t, D) + 1},$$

where ***|D|*** is the total number of documents in the corpus.

Featurization: **Feature Extractors**

- **Term Frequency–Inverse Document Frequency, or TF-IDF,**
 - The product of these values, $TF \times IDF$, shows how relevant a term is to a specific document (i.e., if it is common in that document but rare in the whole corpus).
 - The TF-IDF measure is simply the product of TF and IDF:

$$TFIDF(t, d, D) = TF(t, d) \cdot IDF(t, D).$$

Featurization: **Feature Extractors**

■ Term Frequency–Inverse Document Frequency, or TF-IDF,

Suppose that we have term count tables of a corpus consisting of only two documents, as listed on the right.

Calculate TF-IDF for the term "this".

Document 1

Term	Term Count
this	1
is	1
a	2
sample	1

Document 2

Term	Term Count
this	1
is	1
another	2
example	3

Featurization: **Feature Extractors**

■ TF-IDF (Solution),

Calculating TF for “this”:

$$\text{TF}(\text{“this”}, d1) = 1/5 = 0.2$$

$$\text{TF}(\text{“this”}, d2) = 1/7 = 0.14 \text{ (Approx.)}$$

Document 1

Term	Term Count
this	1
is	1
a	2
sample	1

Document 2

Term	Term Count
this	1
is	1
another	2
example	3

Featurization: **Feature Extractors**

■ TF-IDF (Solution),

**Calculating IDF for
"this":**

$$|D| = 2$$

$$DF(t, D) = 2$$

$$IDF("this", D) = \log \left(\frac{3}{2} \right) = 0$$

$$IDF(t, D) = \log \frac{|D| + 1}{DF(t, D) + 1},$$

Document 1

Term	Term Count
this	1
is	1
a	2
sample	1

Document 2

Term	Term Count
this	1
is	1
another	2
example	3

Featurization: **Feature Extractors**

■ TF-IDF (Solution),

Calculating TF-IDF for “this”:

$$\text{TF-IDF}(\text{“this”}, d1, D) = 0.2 * 0 = 0$$

$$\text{TF-IDF}(\text{“this”}, d2, D) = 0.14 * 0 = 0$$

Document 1

Term	Term Count
this	1
is	1
a	2
sample	1

Document 2

Term	Term Count
this	1
is	1
another	2
example	3

$$TFIDF(t, d, D) = TF(t, d) \cdot IDF(t, D).$$

Featurization: **Feature Extractors**

- **Exercise: Calculate TF-IDF for the term “example”.**

Document 1		Document 2	
Term	Term Count	Term	Term Count
this	1	this	1
is	1	is	1
a	2	another	2
sample	1	example	3

Featurization: **Feature Extractors**

■ **Word2Vec**

- maps each word to a unique fixed-size vector.
- transforms each document into a vector using the average of all words in the document.
- this vector can then be used as features for prediction, **document similarity calculations**, etc.

Home Work: Do some research and write a program to find the document similarity using Word2Vec.

Featurization: **Extraction, transformation and selection**

- **Extraction**

- Extracting features from “raw” data

- **Transformation**

- Scaling, converting, or modifying features

- **Selection**

- Selecting a subset from a larger set of features

Featurization: **Feature Extraction and Transformation**

■ **Feature Transformers**

- Tokenization
- Stop Words Remover
- String Indexing
- One Hot Encoding
- Vector Assembler (Implement In tutorial)

Featurization: **Feature Transformers**

■ Tokenization

- It is the process of taking text (such as a sentence) and breaking it into individual terms (usually words).



Featurization: **Feature Transformers**

- **Stop Words** are words which should be excluded from the input, typically because the words appear frequently and don't carry as much meaning.

Some words contain more information than others

stopwords — [the, in, for, you, will, have, be]

Quiz: How many words will be removed when we remove stopwords from "Hi Katie the machine learning class will be great best Sebastian"

3

Featurization: **Feature Transformers**

■ **Stop Words Remover**

- Takes as input a sequence of strings (e.g. the output of a Tokenizer)
- Drops all the stop words from the input sequences.

id	raw	filtered
0	[I, saw, the, red, balloon]	[saw, red, balloon]
1	[Mary, had, a, little, lamb]	[Mary, little, lamb]

Featurization: **Feature Transformers**

■ String Indexing

- Encoding a string column of labels to a column of label indices.

id	category	categoryIndex
0	a	0.0
1	b	2.0
2	c	1.0
3	a	0.0
4	a	0.0
5	c	1.0

Featurization: Feature Transformers

■ One Hot Encoding

- Maps a categorical feature represented as a label index to a binary vector..
- A single one-value indicates the presence of a specific feature value from among the set of all feature values.
- For string type input data, it is common to encode categorical features using String Indexing first.

Example, three category labels {a,b,c}

In spark, use $N-1$ dimensional binary vector (N is the number of categories)

$$'a' - \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad 'b' - \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad 'c' - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$'a'(index\ 0) - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad 'b'(index\ 1) - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad 'c'(index\ 2) - \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Featurization: **Feature Transformers**

■ **Why One Hot Encoding?**

- For categorical variables when there is no ordinal relationship, the string indexing is not enough...
- Using this encoding and allowing the model to assume a natural ordering between categories may result in poor performance or unexpected results.
- A one-hot encoding can be applied to the integer representation. This is where the integer encoded variable is removed and a new binary variable is added for each unique integer value.

Featurization: **Feature Transformers**

■ Why One Hot Encoding?

- Example: Let's say we have 3 data instances with attributes of Preferred Programming Language and OS of Choice.

Preferred Programming Language	OS of Choice
Javascript	OSX
Python	Linux
Scala	OSX

Featurization: **Feature Transformers**

■ Why One Hot Encoding?

String Indexing

Preferred Programming Language	OS of Choice	Preferred Programming Language	OS of Choice
Javascript	OSX	0	0
Python	Linux	1	1
Scala	OSX	2	0

Featurization: **Feature Transformers**

- **Why cant we STOP here?**
- **The Problem Of Ordinality**
- Machine learning algorithms treat the ordinality of numbers in an attribute with some significance: *a higher number “must be better” than a lower number.*

String Indexing

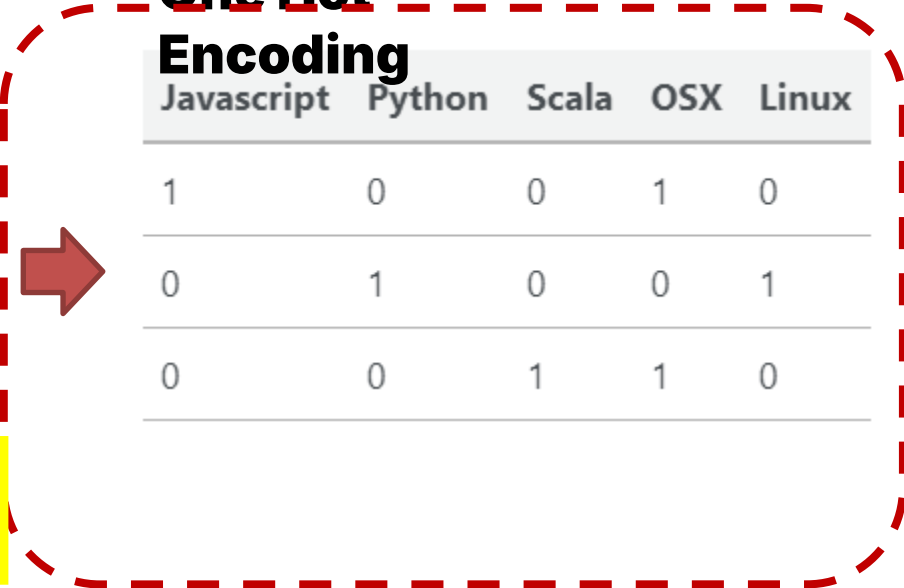
Preferred Programming Language	OS of Choice
0	0
1	1
2	0

Featurization: **Feature Transformers**

- **String Indexing**

Preferred Programming Language	OS of Choice
0	0
1	1
2	0

- **One Hot Encoding**



Javascript	Python	Scala	OSX	Linux
1	0	0	1	0
0	1	0	0	1
0	0	1	1	0

Homework: Have a look at [Principal Component Analysis \(PCA\)](#)

Featurization: **Feature Transformers**

■ **Why One Hot Encoding?**

- For categorical variables when there is no ordinal relationship, the string indexing is not enough...
- Using this encoding and allowing the model to assume a natural ordering between categories may result in poor performance or unexpected results.
- A one-hot encoding can be applied to the integer representation. This is where the integer encoded variable is removed and a new binary variable is added for each unique integer value.

Featurization: **Extraction, transformation and selection**

- **Extraction**

- Extracting features from “raw” data

- **Transformation**

- Scaling, converting, or modifying features

- **Selection**

- Selecting a subset from a larger set of features

Featurization: **Feature Selectors**

■ **Feature selection**

- This process tries to get most important features that are contributing to decide the label.

■ **Vector Slicer**

- It takes a feature vector and outputs a new feature vector with a sub-array of the original features.
- It is useful for extracting features from a vector column.

userFeatures		features
-----		-----
[0.0, 10.0, 0.5]		[10.0, 0.5]

Lecture Demo

■ Spam Classification Example

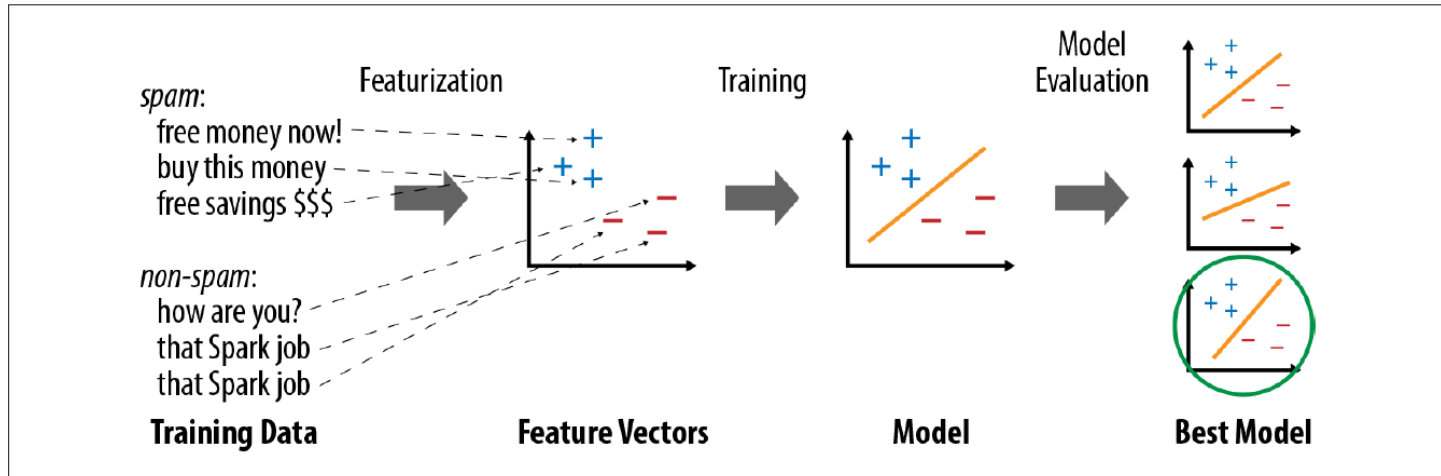


Figure 11-1. Typical steps in a machine learning pipeline

Thank You

See you next week