



ML.js Crash Course

Crashing Machine Learning w/ JavaScript

 @joshuamabina

What is machine learning?



@joshuamabina

Artificial intelligence,
Machine learning,
Data science,
Big data

??



@joshuamabina



About me

Developer



Vim



Data-science



 @joshuamabina

Worth mentioning...

1. Preach JavaScript
2. #ALCinTanzania
3. Comunity Ninja
4. Marathoner



@joshuamabina



https://medium.com/@forlooptanzania_99946/forloop-tanzania-major-updates-7ba32ebe3190

#ALCwithforLoop

 @joshuamabina

Can Digital Images Talk? How can ML & JS Help?



<https://github.com/joshuamabina/claws>

 @joshuamabina

What is machine learning??



@joshuamabina

Machine learning...

Machine learning (ML) is the **study of algorithms** and **mathematical models** that computer systems use to **progressively improve their performance on a specific task.**

~ Wikipedia

Machine learning...

Machine learning algorithms **build a mathematical model of sample data**, known as "training data", in order to **make predictions or decisions without being explicitly programmed** to perform the task.

~ Wikipedia

What are the common
machine learning problems?



@joshuamabina

Supervised learning, a type of machine learning, can be approached through classification.

Classification is a process in which a computer learns to identify to which class the given input belongs to by learning from labelled data.

Simply put,

1. It is a subfield of Artificial intelligence (AI)
2. It is a study of algorithms that learn from examples and experience
3. It is a study of algorithms that help computers learn without being explicitly programmed to do so
4. It is a study of algorithms that help computers predict correct answers & execute instructions without your explicit directions

What is the connection between artificial intelligence, machine learning, deep learning, data-science and big-data?

#foodForThought

 @joshuamabina

Head first into machine learning...

Problem: Is it a Mouse or Rat?! 🤔



@joshuamabina

The steps

1. Import your data set
2. Train your model
3. Predict the correct answer

The rodents data...

Weight (units)	Location	Label
140	Town (0)	Mouse (0)
120	Town (0)	Mouse (0)
700	Country (1)	Rat (1)
500	Country (1)	Rat (1)
900	Country (1)	Rat(1)
100	Town (0)	Mouse (0)

Importing the data:

```
features = [[140, 0], [120, 0], ..., [500, 1] [900, 1]]
```

```
labels = [0, 0, 1, 1, 1, 0]
```


Training the classifier:

```
classifier = tree.DecisionTreeClassifier()
```

```
classifier = classifier.fit(features,label)
```



@joshuamabina

Predict the rodent:

```
print classifier.predict([[300, 0]])
```

```
from sklearn import tree
```

```
# 1. import the data
```

```
features = [[140, 0], [120, 0], [700, 1], [500, 1], [900, 1], [100, 0]]
```

```
label = [0, 0, 1, 1, 1, 0]
```

```
# 2. train the classifier
```

```
classifier = tree.DecisionTreeClassifier()
```

```
classifier = classifier.fit(features, label)
```

```
# 3. predict
```

```
print classifier.predict([[300, 0]])
```

How accurate will the classifier be when used on rodents not in the training data?



@joshuamabina

One way to evaluate an algorithm is by partitioning it into two: (1) the training set, on which we learn some properties; (2) the testing set, on which we test the learned properties.

```
from sklearn import tree
```

```
# 1. import the data
```

```
testing_features = [[140, 0], [700, 1]]
```

```
testing_labels = [0, 1]
```

```
training_features = [[120, 0], [500, 1], [900, 1], [100, 0]]
```

```
training_labels = [0, 1, 1, 0]
```

```
# 2. train the classifier
```

```
classifier = tree.DecisionTreeClassifier()
```

```
classifier = classifier.fit(training_features, training_labels)
```

```
# 3. predict
```

```
print classifier.predict(testing_features)
```

```
print testing_labels
```




It's JavaScript

<https://github.com/getify/You-Dont-Know-JS>

JavaScript Ecosystem?



1. Node, npm, nvm

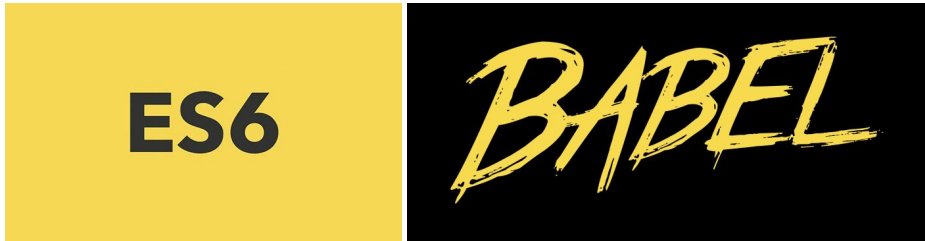
2. ES6 & Babel



3. Webpack



webpack
MODULE BUNDLER

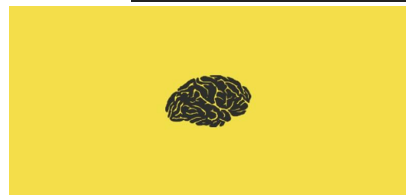


<https://github.com/creationix/nvm>

 @joshuamabina

JavaScript ML Ecosystem?

1. brain.js (Neural Networks)
2. Synaptic (Neural Networks)
3. Natural (Natural Language Processing)
4. ConvNetJS (Convolutional Neural Networks)
5. **mljs (A set of sub-libraries with a variety of functions)**
6. Neataptic (Neural Networks)
7. Webdnn (Deep Learning)
8. Tensorflow.js (Deep Learning)





ml.js

Machine learning and numerical analysis tools in JavaScript for Node.js and the Browser !

[Report abuse](#)

 **Repositories** 85

 **People** 7

 **Projects** 1

Pinned repositories



ml

Machine learning tools in JavaScript

 JavaScript ★ 1.3k 🍴 136



matrix

Matrix manipulation and computation library

 JavaScript ★ 108 🍴 16



pca

Principal component analysis

 JavaScript ★ 50 🍴 11



knn

A k-nearest neighbor classifier algorithm.

 JavaScript ★ 59 🍴 10



feedforward-neural-networks


A implementation of feedforward neural networks based on wildml implementation

 JavaScript ★ 17 🍴 3



libsvm

LIBSVM for the browser and nodejs 🔥

 JavaScript ★ 37 🍴 7

<https://github.com/mljs>



@joshuamabina

Importing the data:

```
let trainingSet = [[140, 0], [120, 0], [700, 1], [500, 1], [900, 1], [100, 0]]  
    let trainingLabels = [0, 0, 1, 1, 1, 0];
```



@joshuamabina

```
import { DecisionTreeClassifier as DTClassifier } from 'ml-cart';
```

```
// 1. import the data
```

```
let trainingSet = [[140, 0], [120, 0], [700, 1], [500, 1], [900, 1], [100, 0]]
```

```
let trainingLabels = [0, 0, 1, 1, 1, 0];
```

```
// 2. train the classifier
```

```
let classifier = new DTClassifier();
```

```
classifier.train(trainingSet, trainingLabels);
```

```
// 3. predict the classifier
```

```
console.log(classifier.predict([[300, 0]]));
```

Recap / Q&A