# AI WORKSHOP

## How we're using Data Science at Groupon

Adam, Aleksander, Roksana and Sviatoslav

Katowice, 12.03.2019

GROUPON

# Agenda

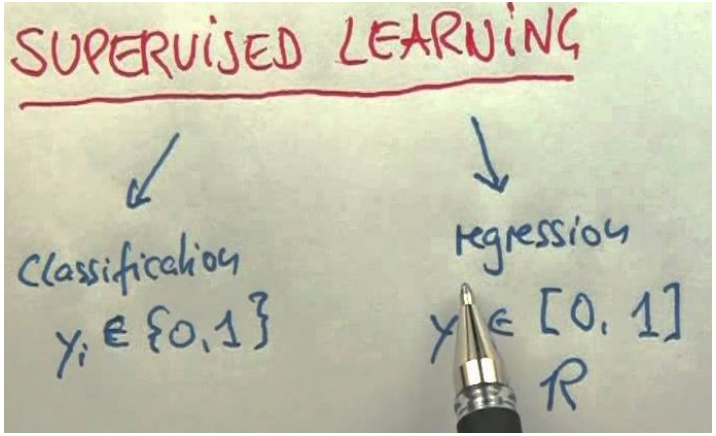
- ★ Groupon - what we’re doing
- ★ Engineering at Groupon
- ★ WORKSHOP
- ★ Real life example
- ★ Summary

GROUPON®

# Part 1

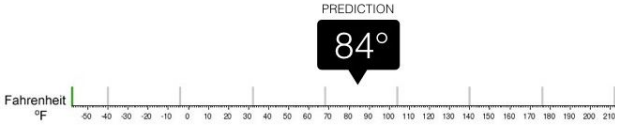
## Standard Machine Learning

# Machine Learning



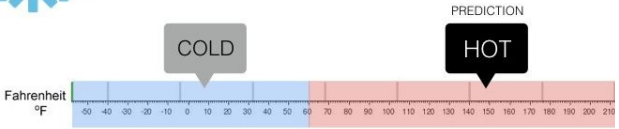

**Regression**

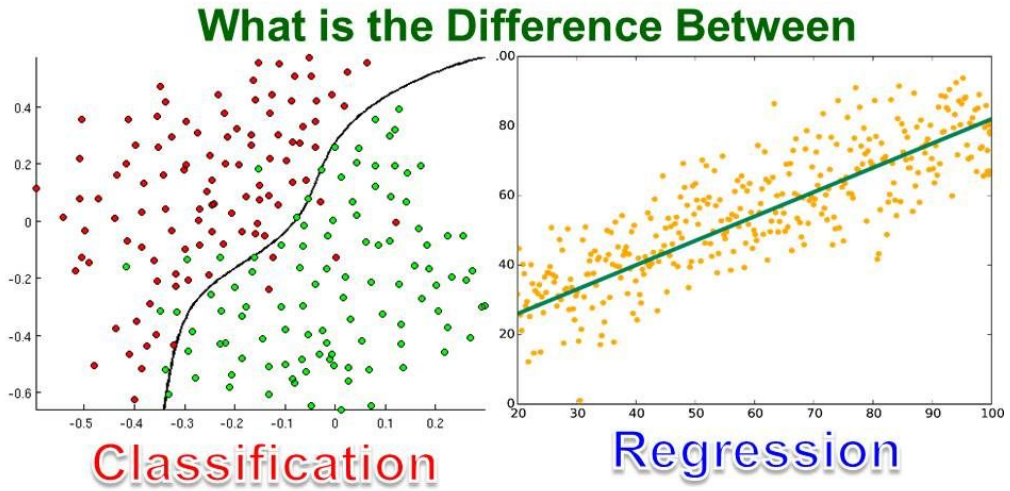
What is the temperature going t
be tomorrow?




**Classification**

Will it be Cold or Hot tomorrow?



# Run Machine Learning Models

## Laptop

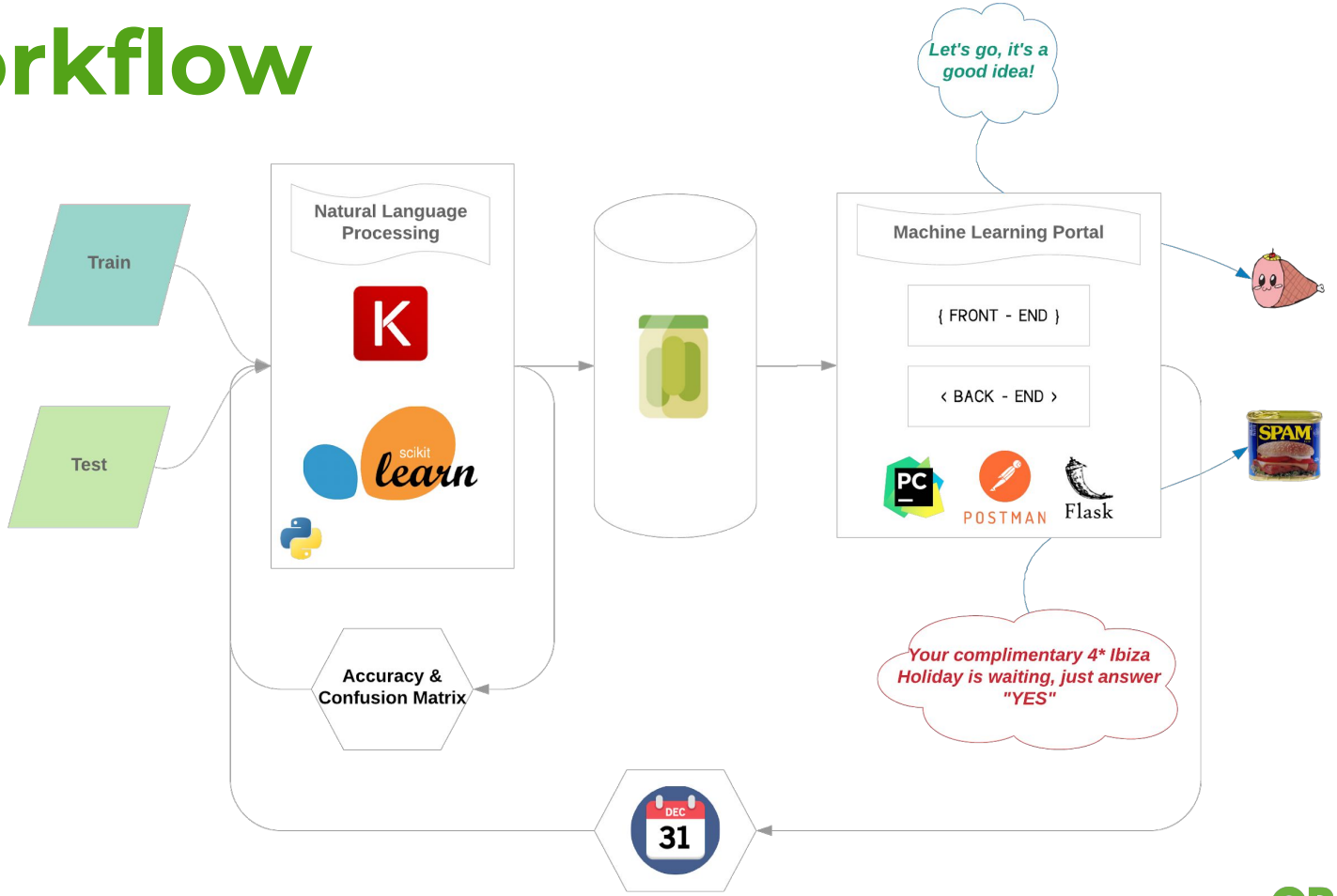

## Cloud

kaggle.com



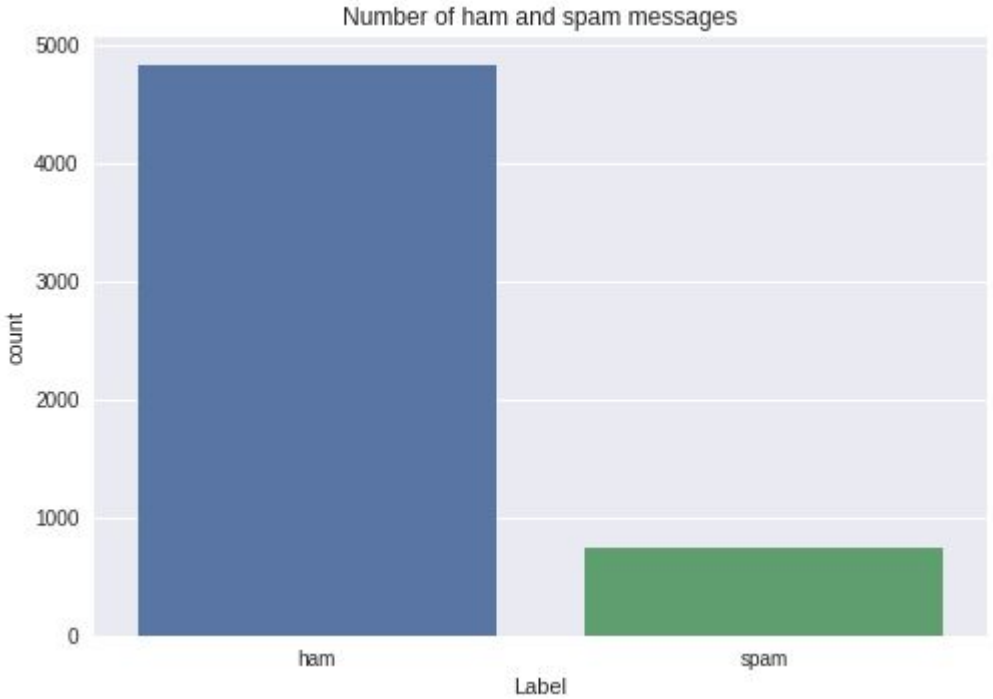
GROUPON

# Workflow




GROUPON

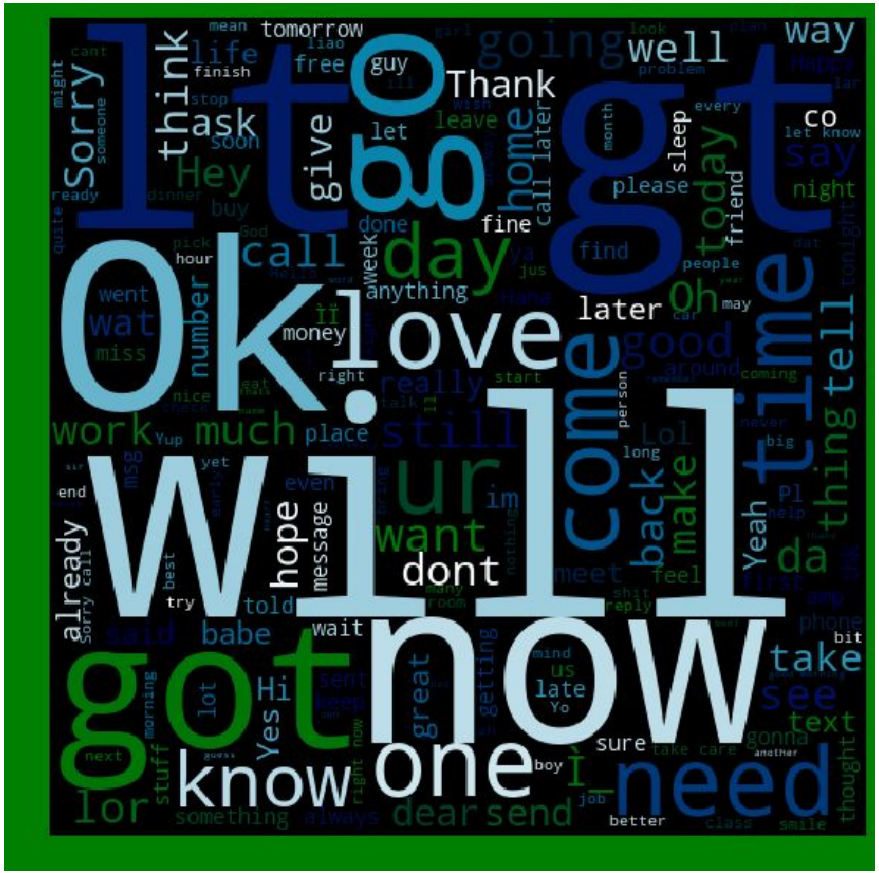
# SPAM vs HAM - sms textual data



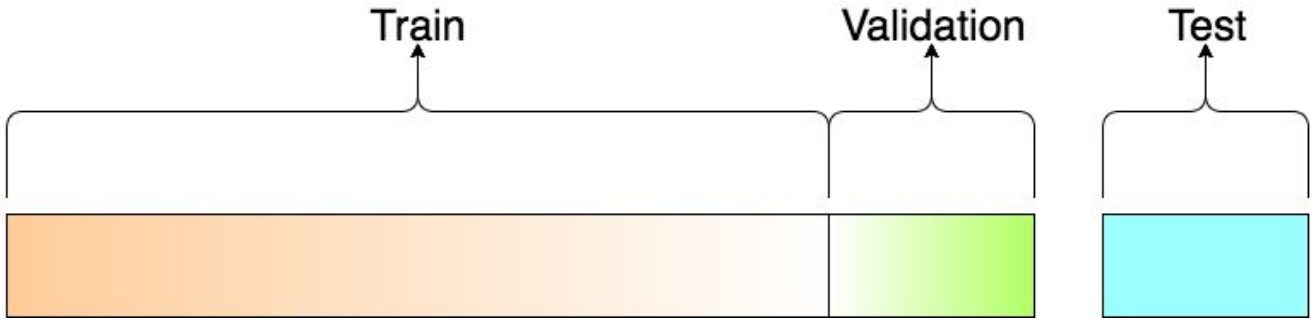

| | Label | Text |
|---|---|---|
| 0 | ham | Go until jurong point, crazy.. Available only ... |
| 1 | ham | Ok lar... Joking wif u oni... |
| 2 | spam | Free entry in 2 a wkly comp to win FA Cup fina... |
| 3 | ham | U dun say so early hor... U c already then say... |
| 4 | ham | Nah I don't think he goes to usf, he lives aro... |
| 5 | spam | FreeMsg Hey there darling it's been 3 week's n... |
| 6 | ham | Even my brother is not like to speak with me. ... |
| 7 | ham | As per your request 'Melle Melle (Oru Minnamin... |
| 8 | spam | WINNER!! As a valued network customer you have... |
| 9 | spam | Had your mobile 11 months or more? U R entitle... |

GROUPON

# SPAM vs HAM - exploratory analysis

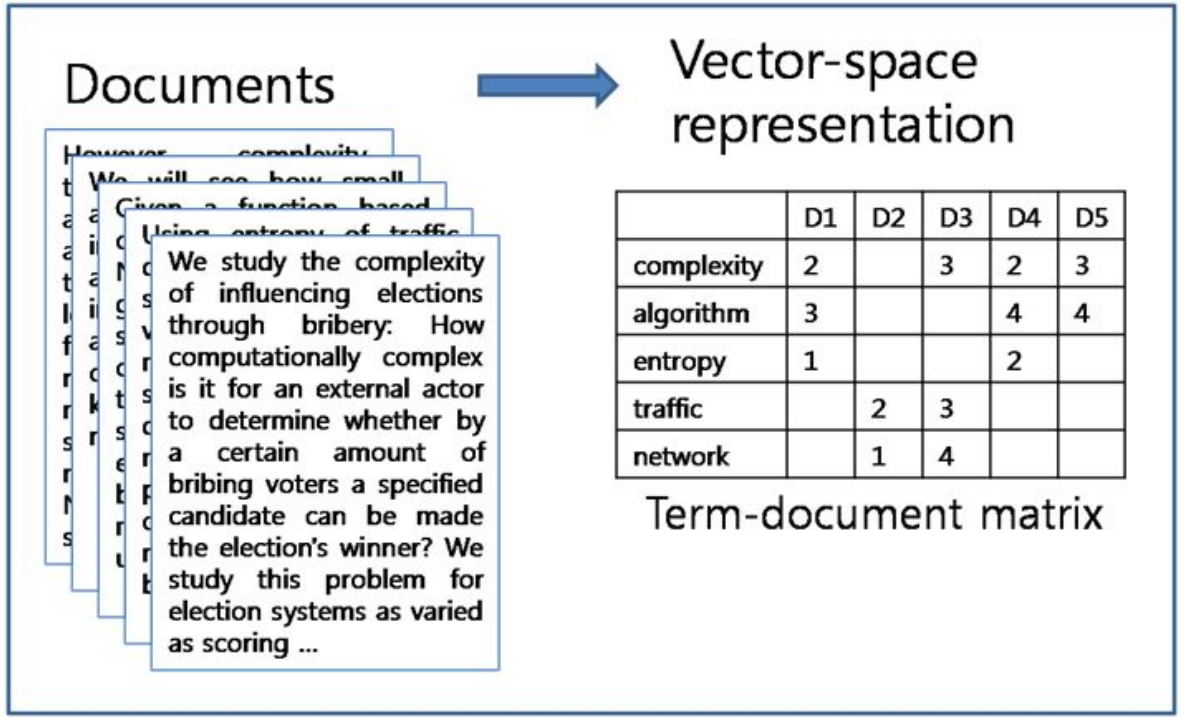



# Train / Test Split


GROUPON

# CountVectorizer creates DTM Document Term Matrix

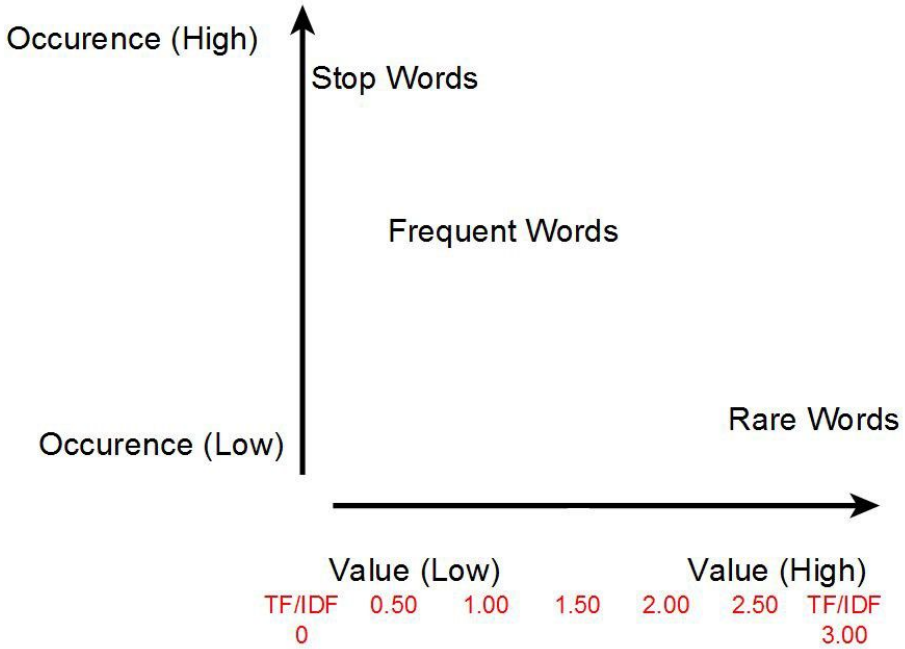


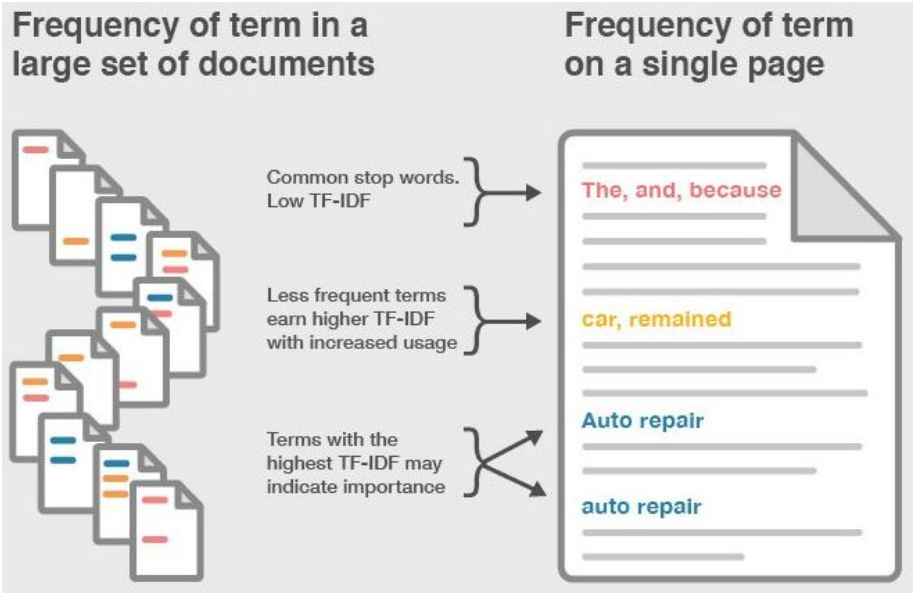

| | D1 | D2 | D3 | D4 | D5 |
|---|---|---|---|---|---|
| complexity | 2 | | 3 | 2 | 3 |
| algorithm | 3 | | | 4 | 4 |
| entropy | 1 | | | 2 | |
| traffic | | 2 | 3 | | |
| network | | 1 | 4 | | |

Term-document matrix

GROUPON

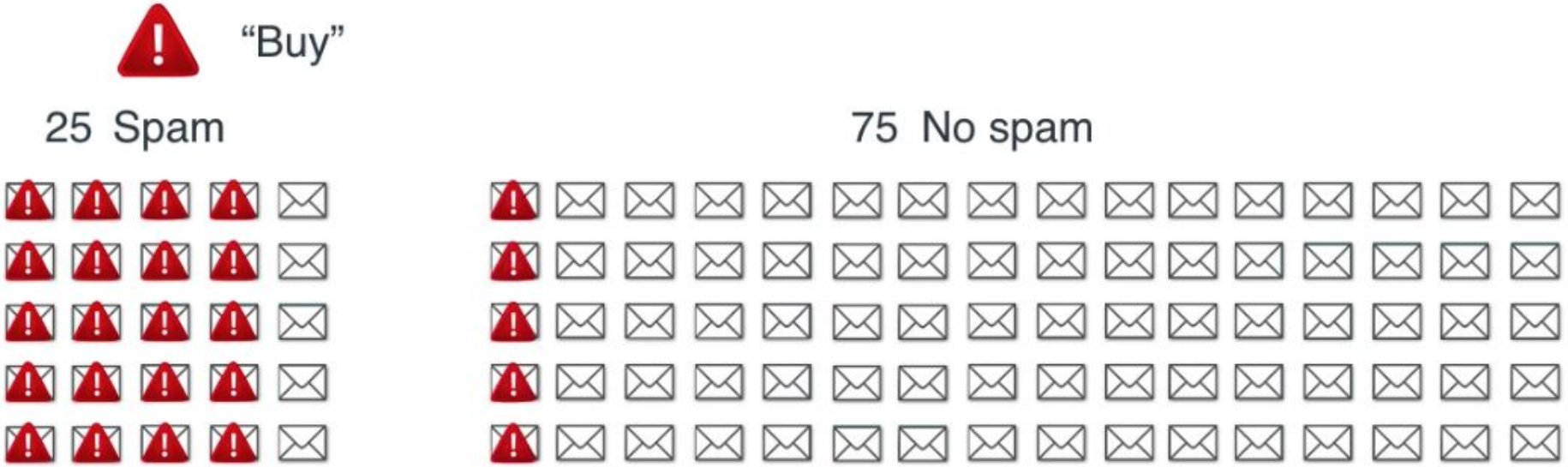
# TF-IDF Term Frequency Inverse Document Frequency




GROUPON

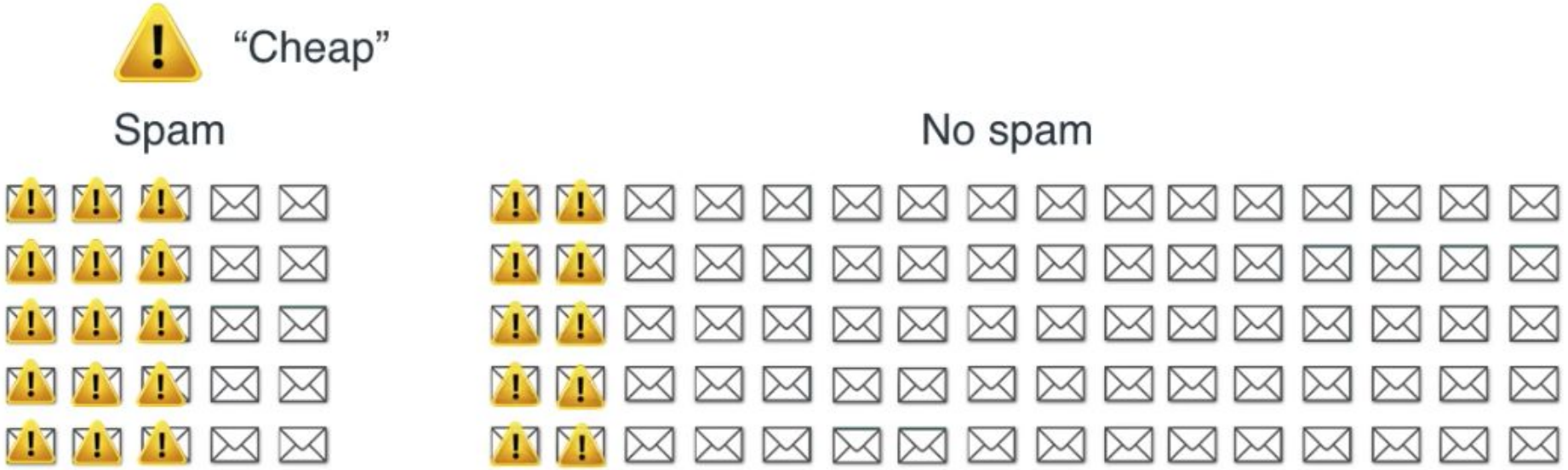
# NB - Naive Bayes (1)




**P(“Buy” is Spam) = ?**

GROUPON

# NB - Naive Bayes (2)




**P(“Cheap” is Spam) = ?**

GROUPON

# NB - Naive Bayes (3)

"Buy" and "Cheap"

Spam

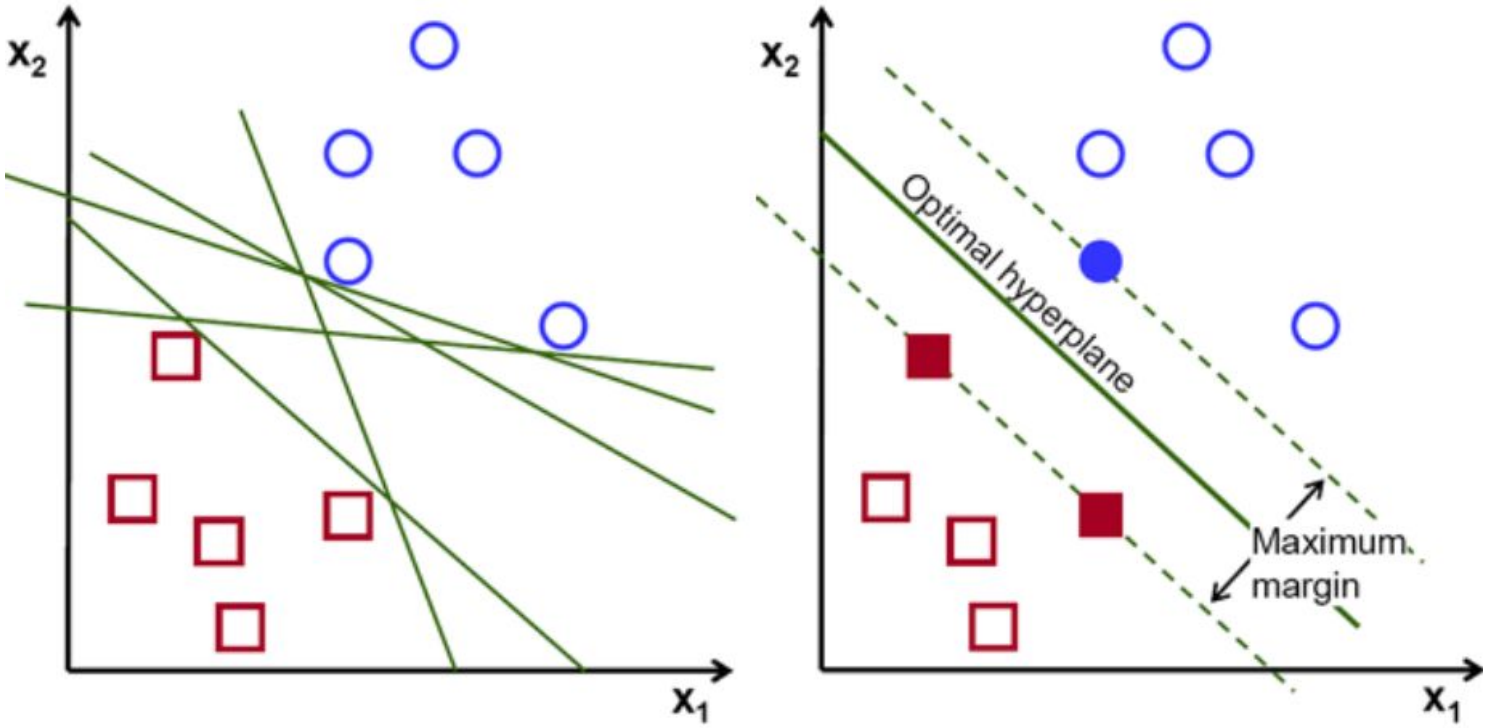
No spam

**P("Buy" & "Cheap" is Spam) = ?**

Naive Bayes classifier: A friendly approach
https://www.youtube.com/watch?v=Q8l0Vip5YUw

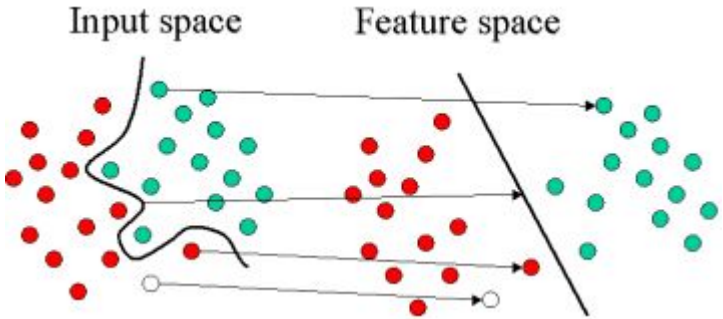
GROUPON

# SVM - Support Vector Machine



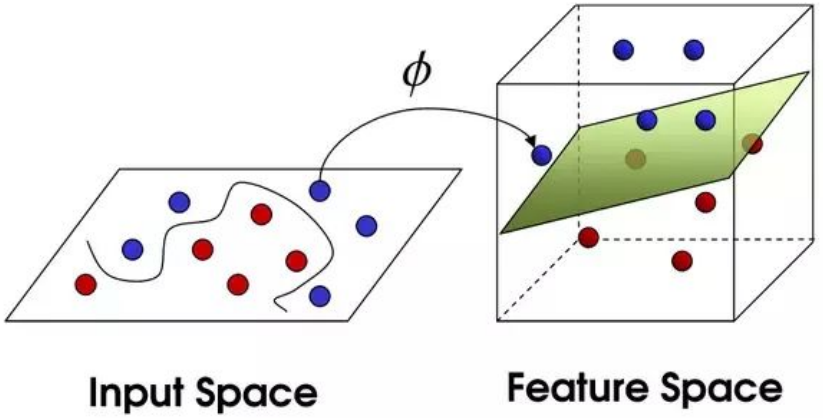

GROUPON

# SVM - Tuning Hyper Parameters

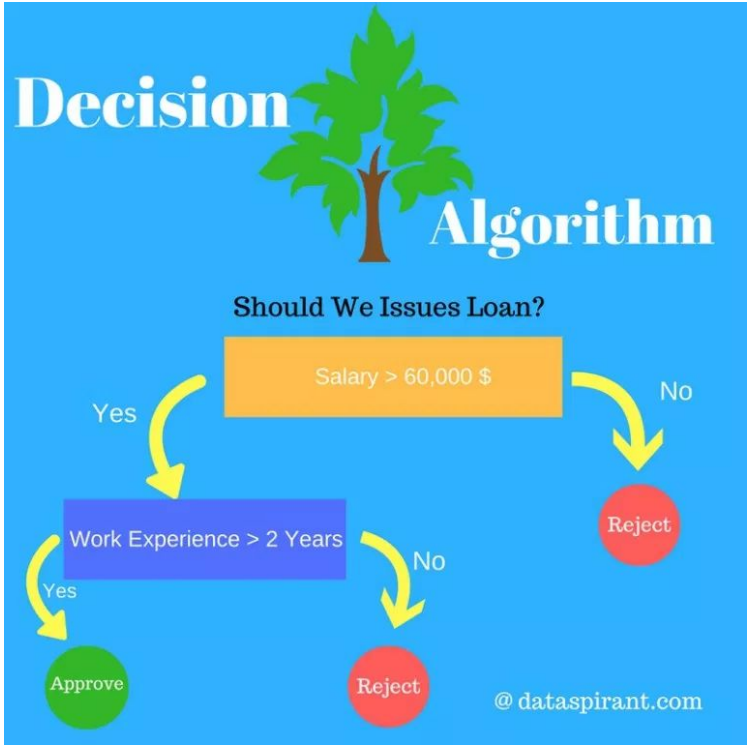


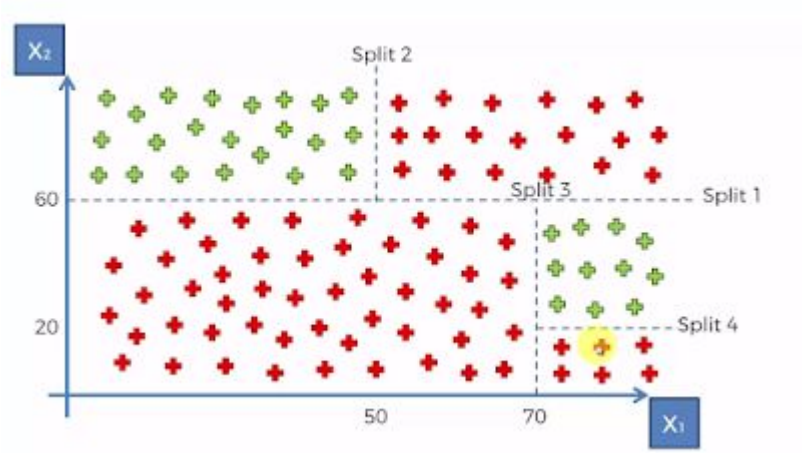

## SVM Kernel

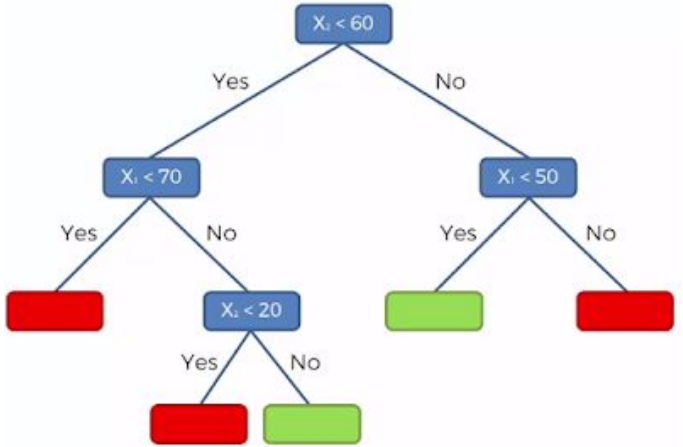
**$\phi$ - is mapping between Non-Linear and Linear Separable**



# DT - Decision Tree

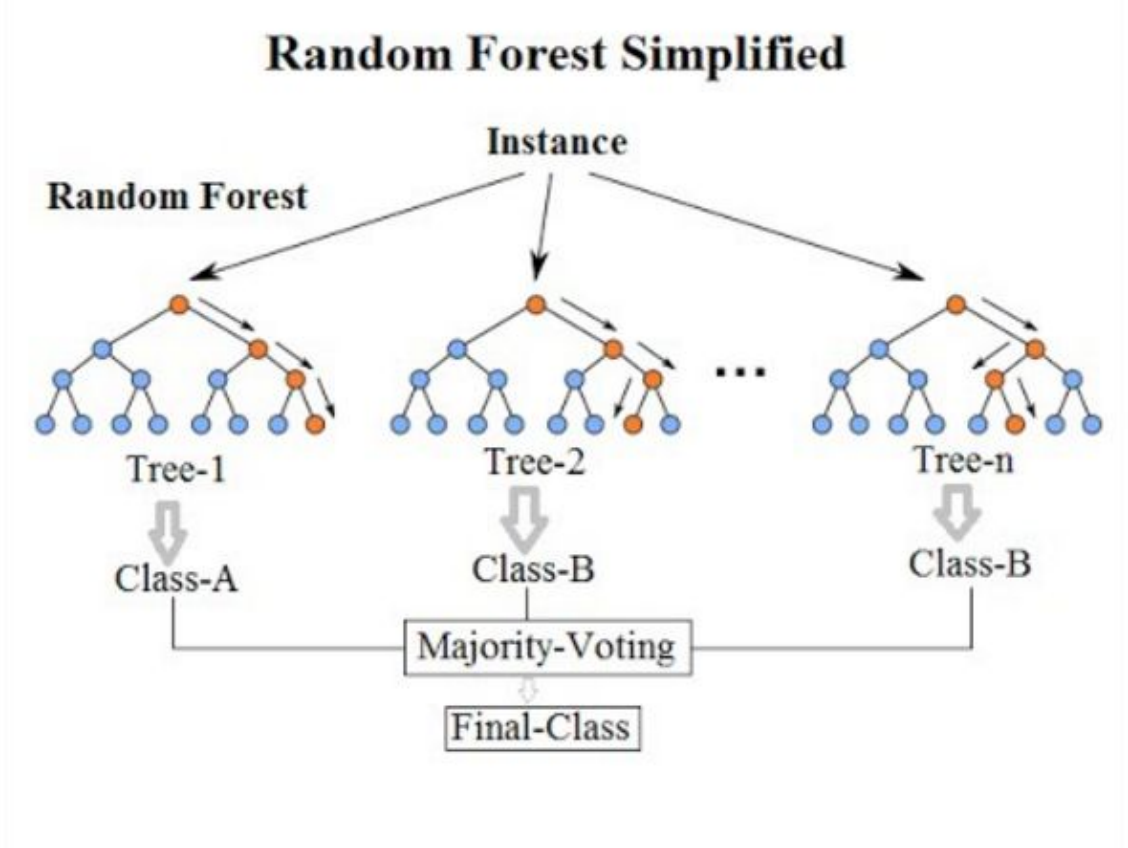



GROUPON

# Random Forest - many DT and Voting



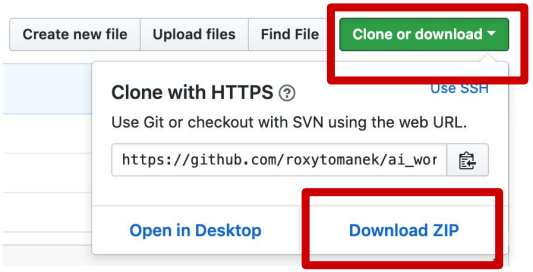

GROUPON

# WORKSHOP

## Automatic Document Categorization with Machine Learning techniques in Python

https://github.com/roxytomanek/ai_workshop (All data and information)

1) Go to the github page and download files




2) Unpack the file
3) Start the Jupyter Notebook
4) Run file
   *'./Workshop/Standard NLP Approach.ipynb'*

GROUPON

# colab.research.google.com

Notebook Address https://colab.research.google.com/drive/1mC8xs1J7nGlpDbCt2oCuFcsfGJ545QiM


# Upload file - Google Colab Session

1

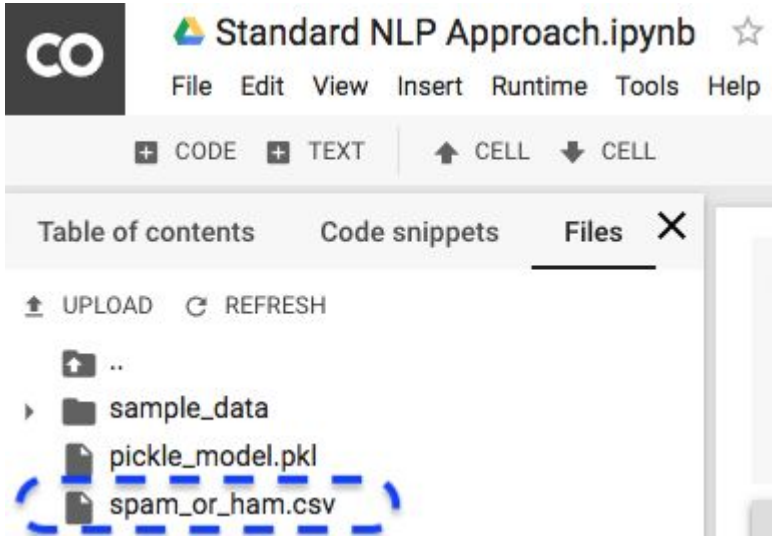
```
# Upload File to Gooogle Colab
from google.colab import files

uploaded = files.upload()
```

2

Wybierz pliki | Nie wybrano pliku  Upload widget is only

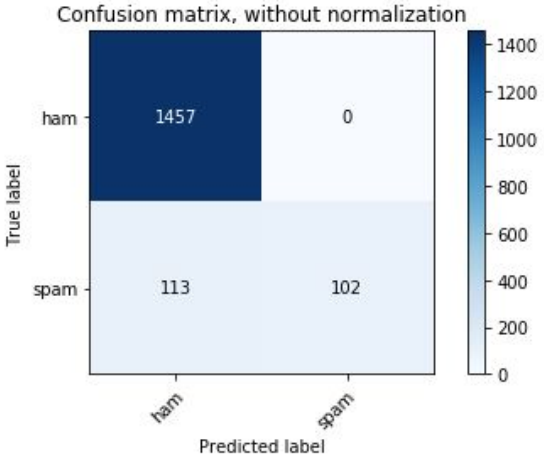
```
Saving spam_or_ham.csv to spam_or_ham.csv
```

3




GROUPON

# Task to do...
# Accuracy & Computation Time




Naive Bayes

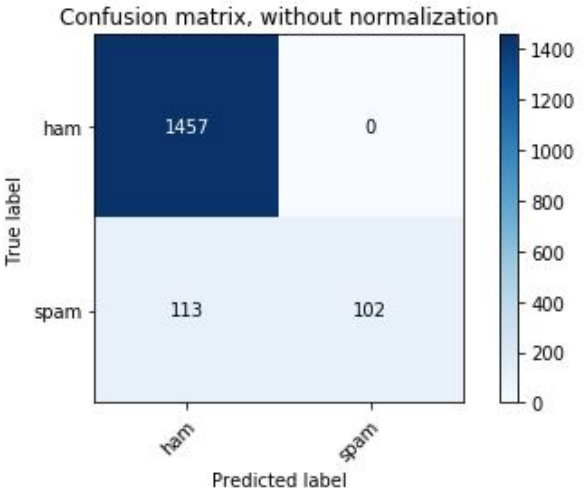
?

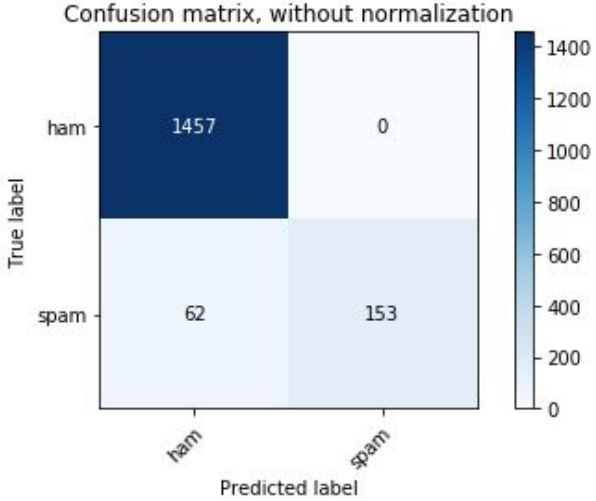
Random Forest

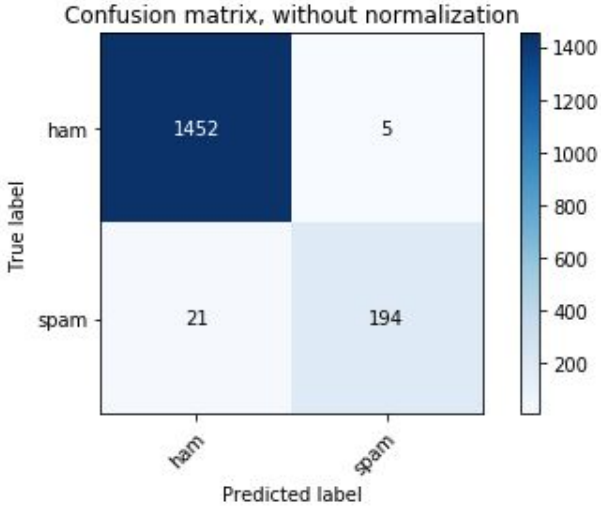
?

Support Vector Machine

GROUPON




# NB

```
Training done in 0.366s
Model final score: 0.932
```

# RF

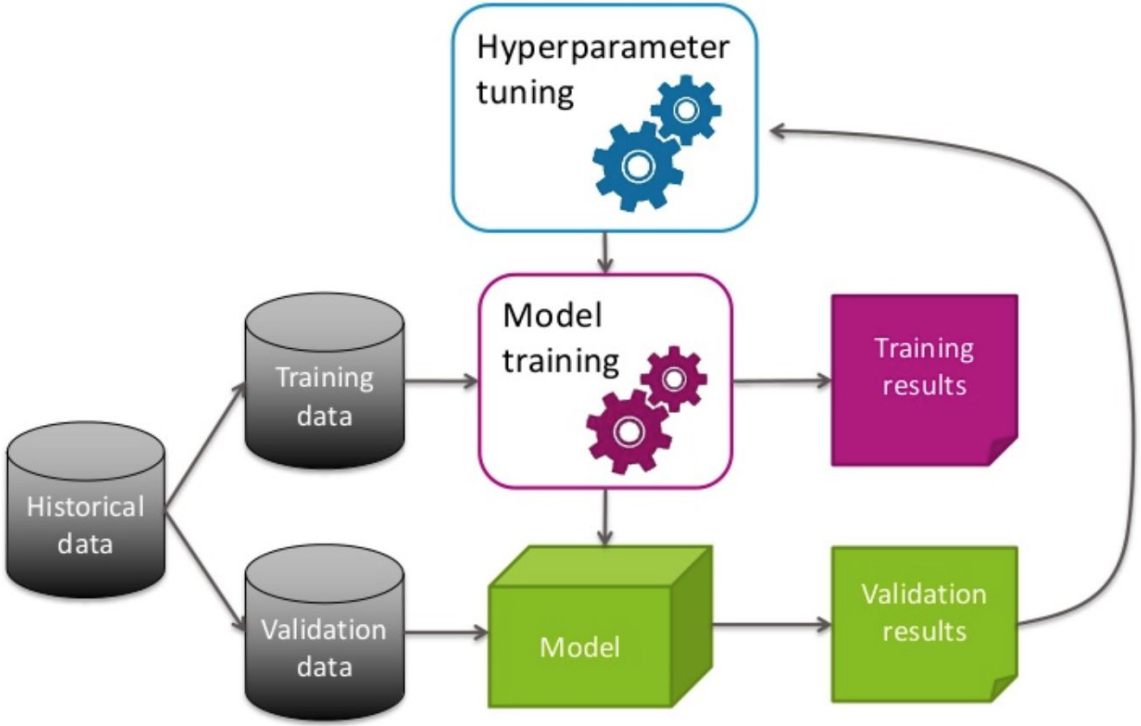
```
Training done in 0.480s
Model final score: 0.963
```

# SVM

```
Training done in 5.247s
Model final score: 0.984
```

GROUPON

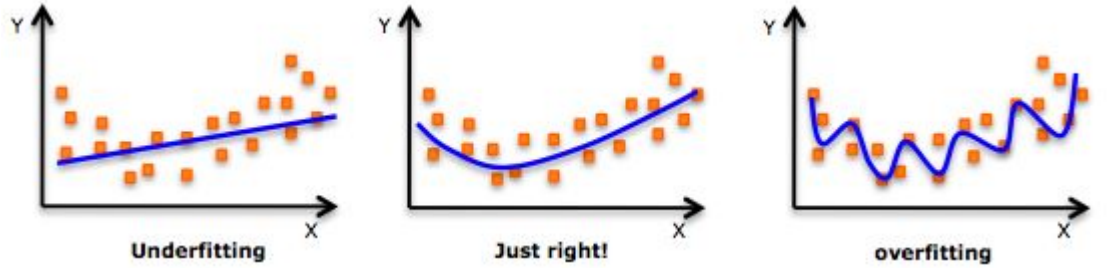
# Model Selection and Tuning

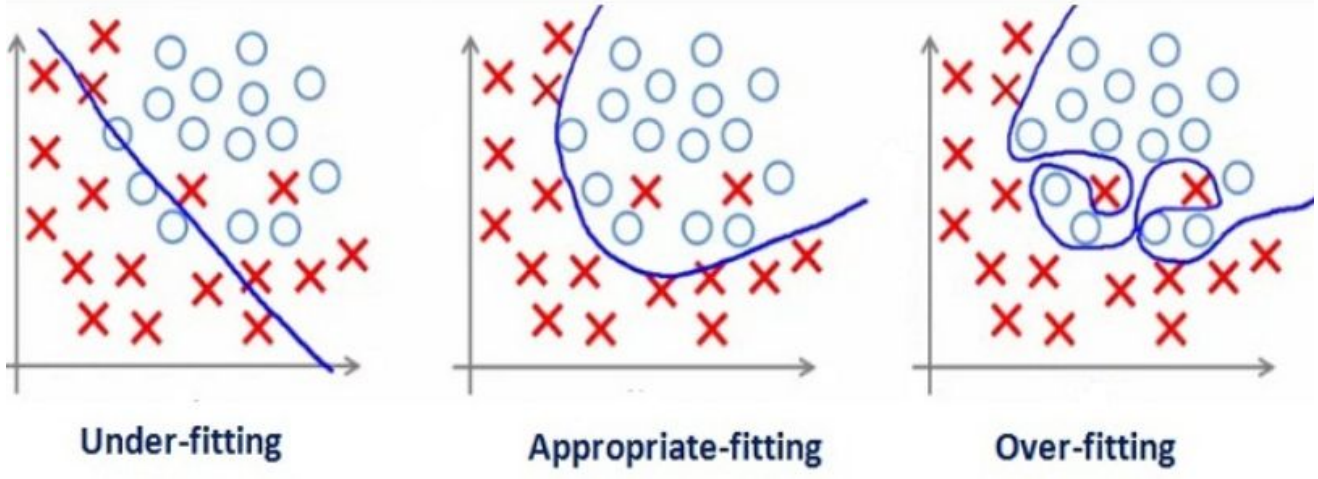



https://www.slideshare.net/AliceZheng3/evaluating-machine-learning-models-a-beginners-guide

GROUPON

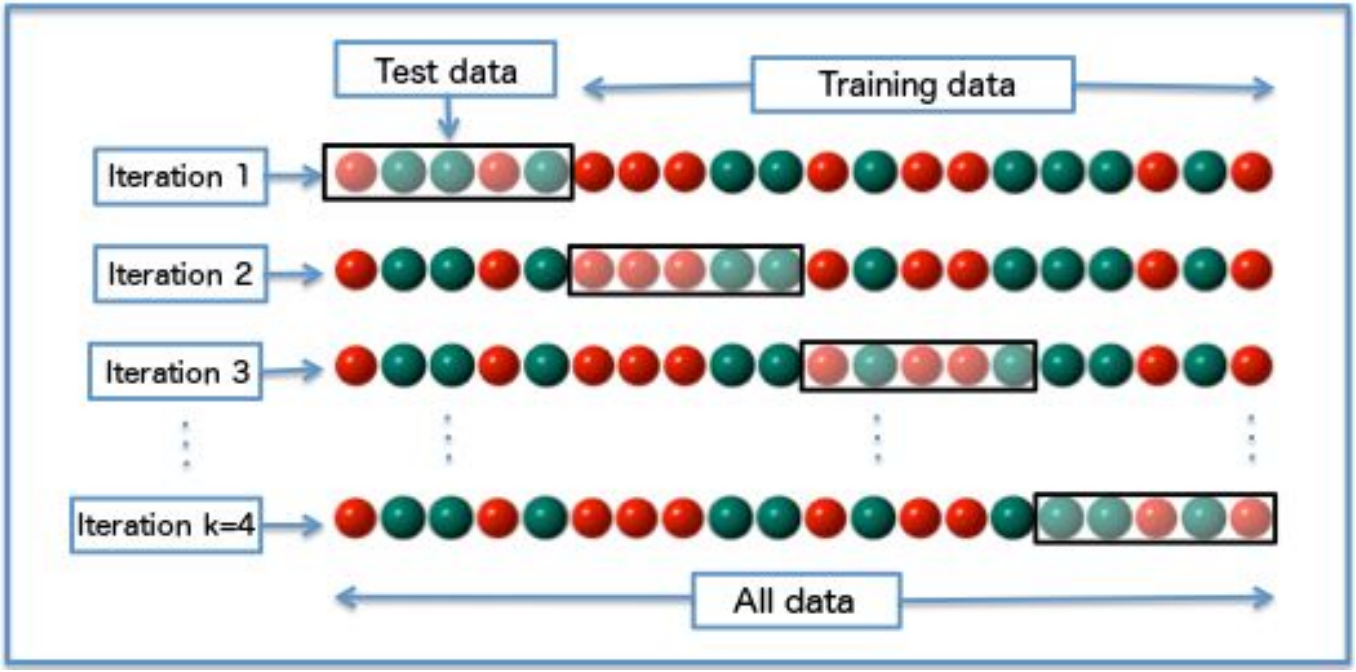
# Underfit and Overfit



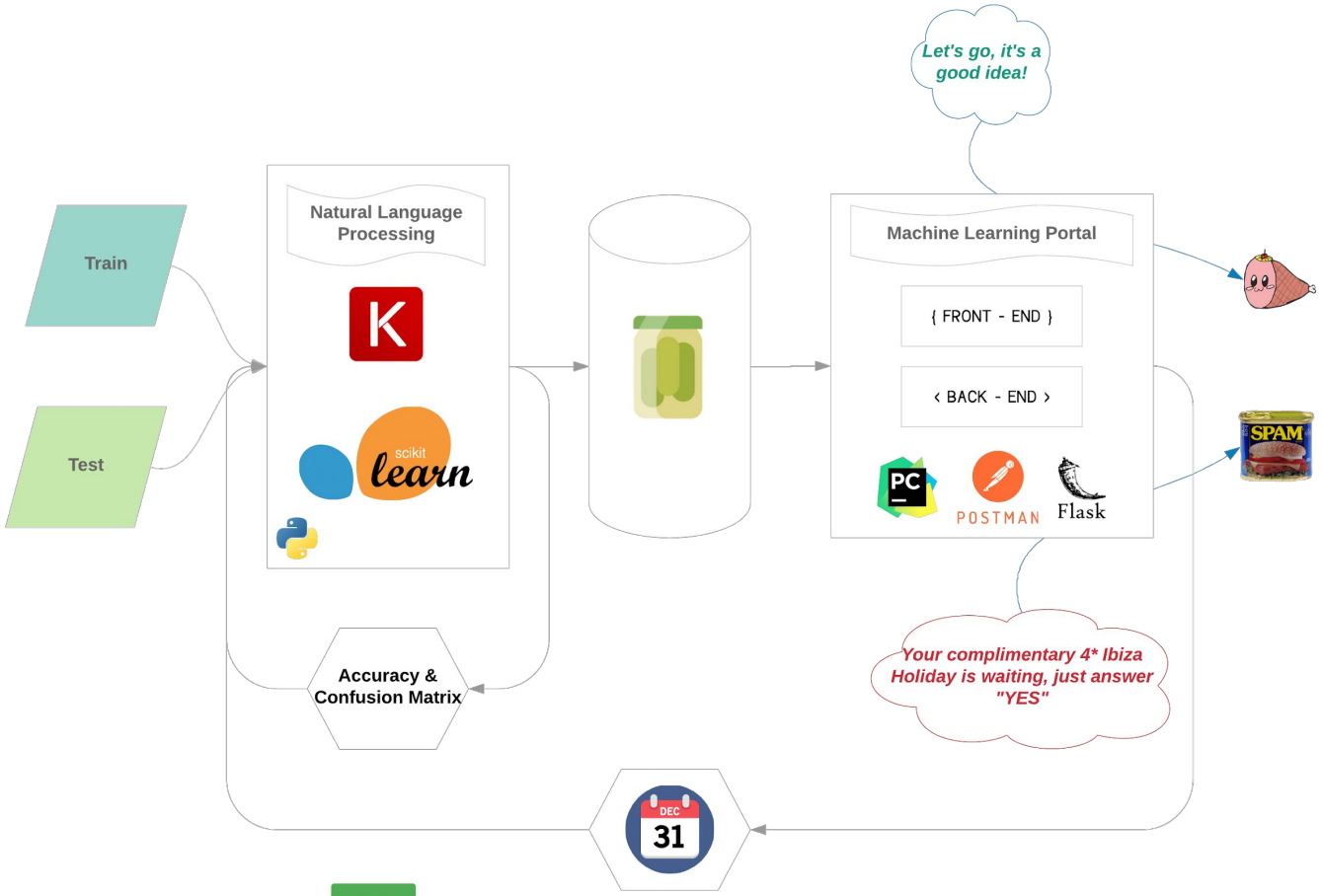

GROUPON

# Against Overfit - Cross-Validation




## 4-Fold Cross-Validation

GROUPON




# List of engines

Search

ADD

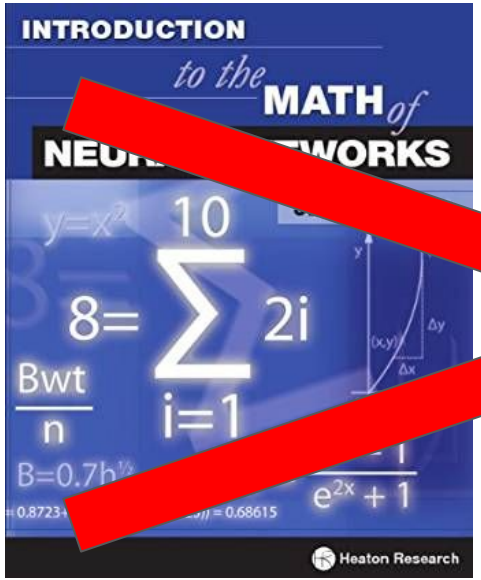
| # | Name | Accuracy | Samples | |
|---|---|---|---|---|
| 1 | spam_ham | 0.98 | 5572 | ✖ |

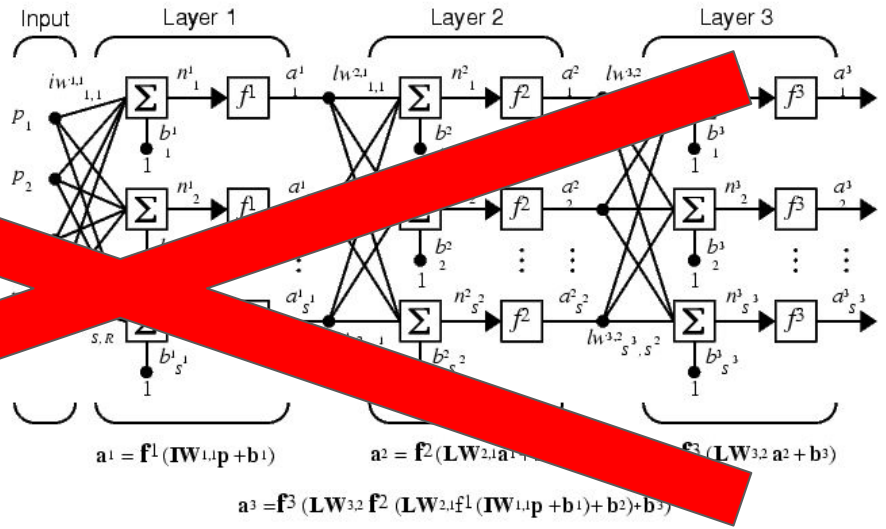
GROUPON

# Part 2

## Neural Networks

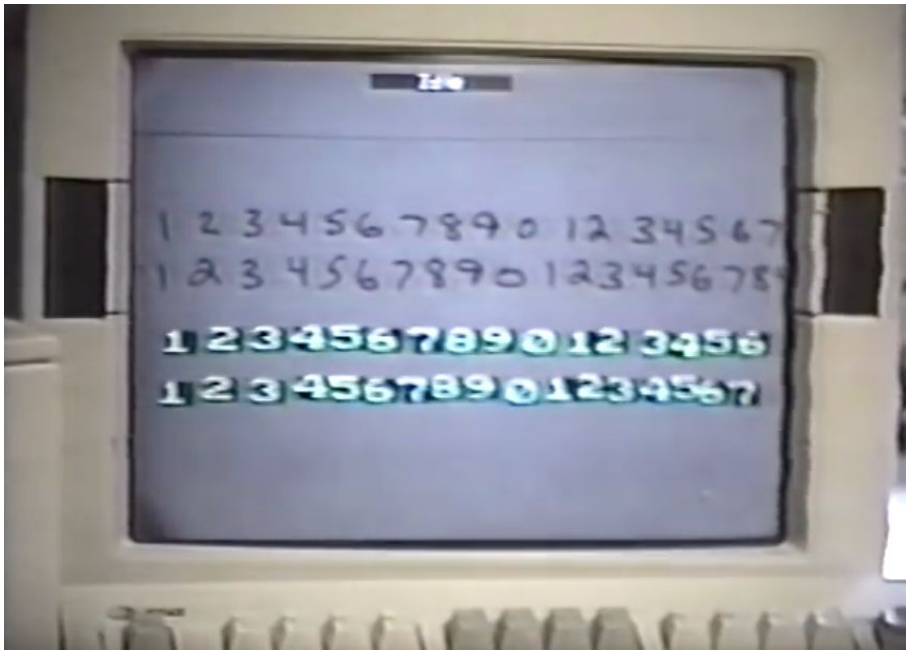
# Neural Networks University Approach

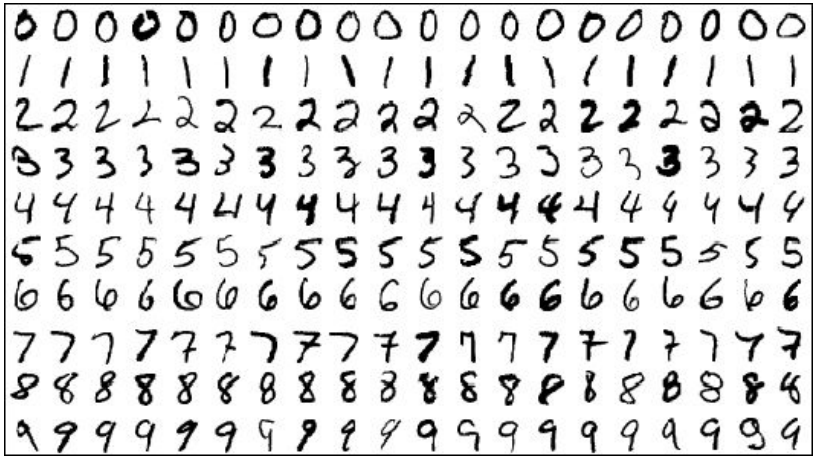



GROUPON

# MNIST - Digit Recognition



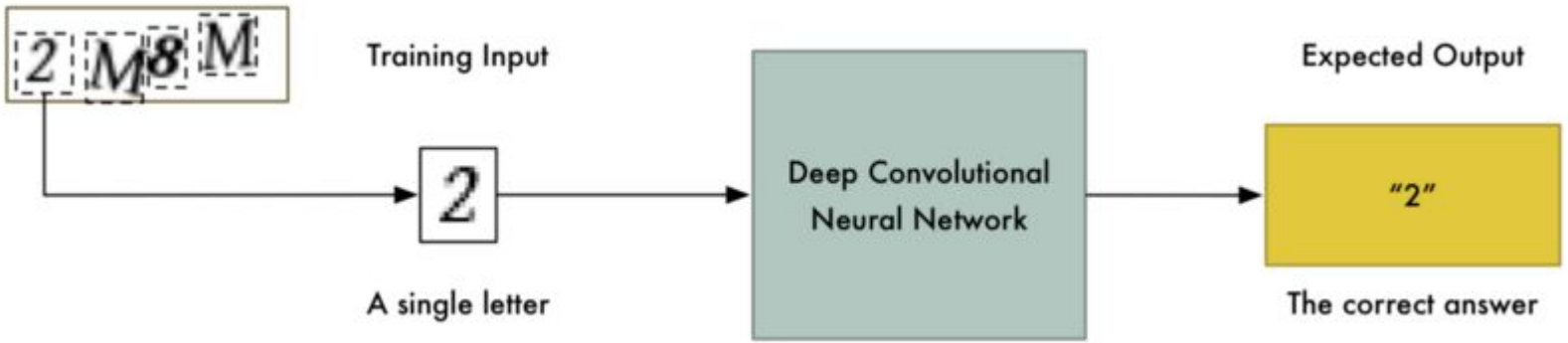


## Convolutional Network Demo from 1993

https://www.youtube.com/watch?v=FwFduRA_L6Q

GROUPON

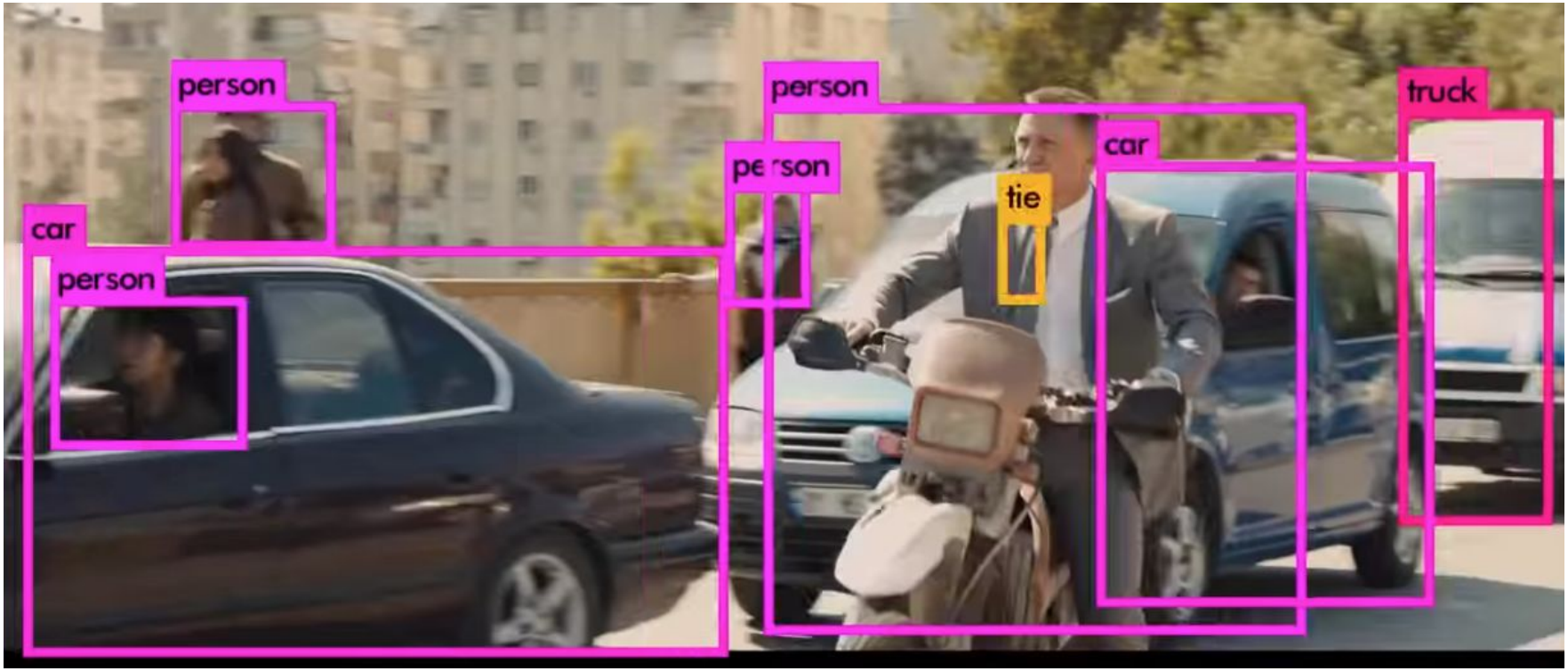
# How to break a CAPTCHA system in 15 minutes with Machine Learning




https://medium.com/@ageitgey/how-to-break-a-captcha-system-in-15-minutes-with-machine-learning-dbebb035a710

GROUPON

# Why Neural Networks ?

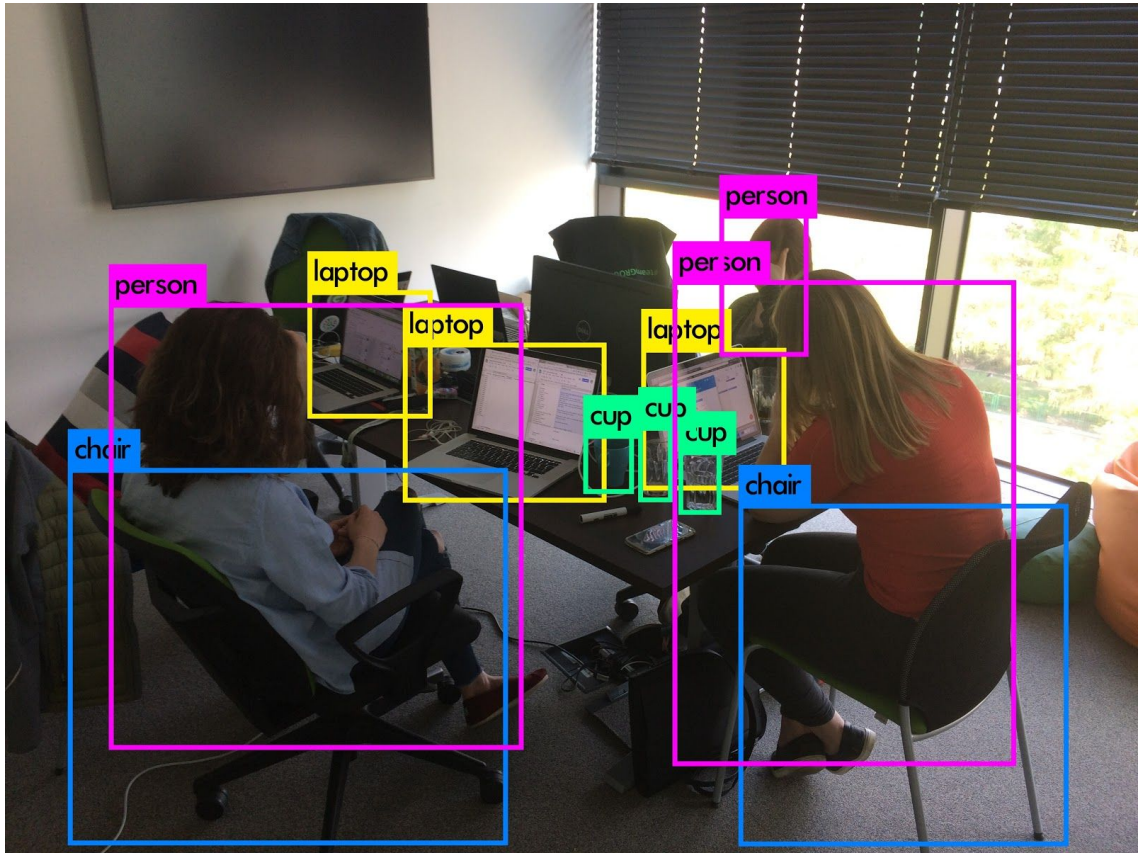



Yolo ver 2 https://www.youtube.com/watch?v=VOC3huqHrss

TED | Joseph Redmon https://www.youtube.com/watch?v=CgxsvIriJhI

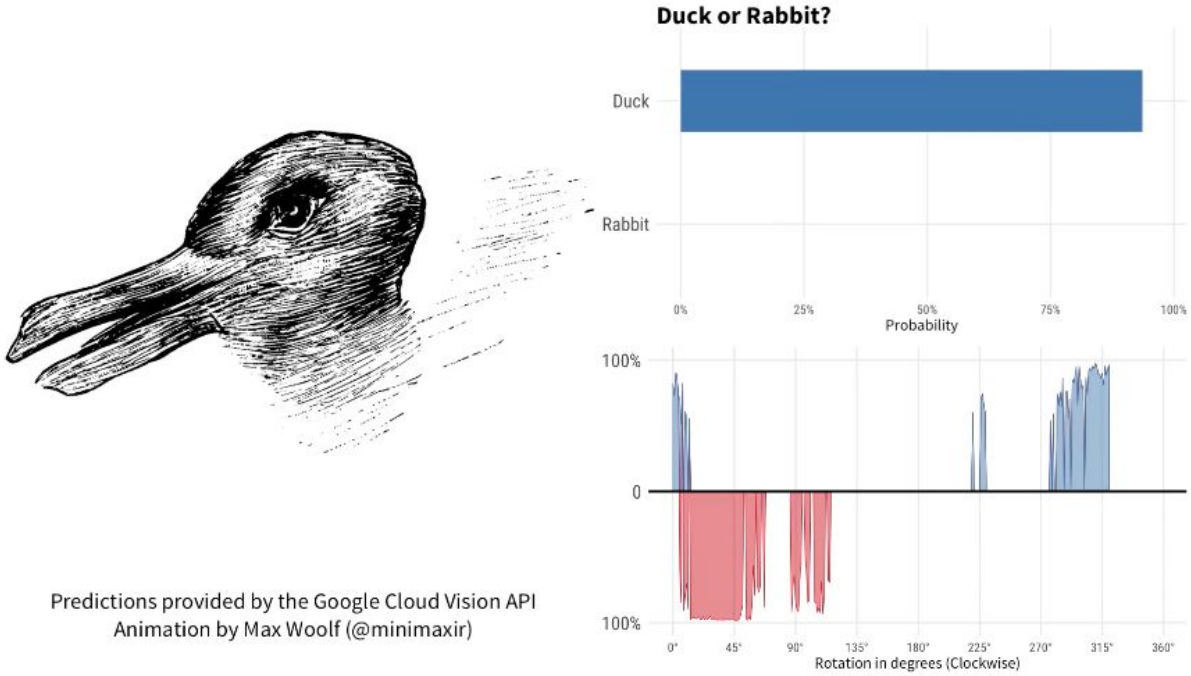
GROUPON

# YOLO in Groupon Katowice




GROUPON

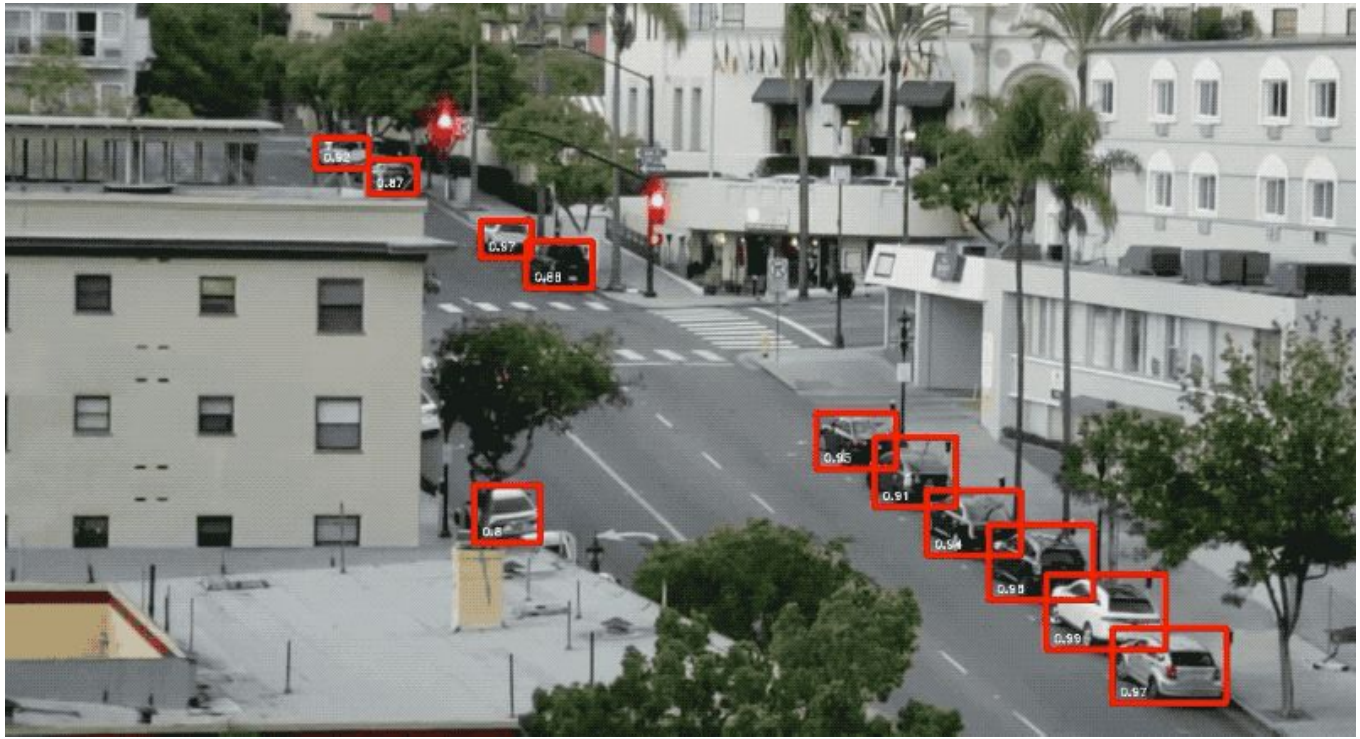
# Hacking CNN, Duck or Rabbit




https://www.reddit.com/r/dataisbeautiful/comments/aydqig/is_it_a_duck_or_a_rabbit_for_google_cloud_vision

GROUPON

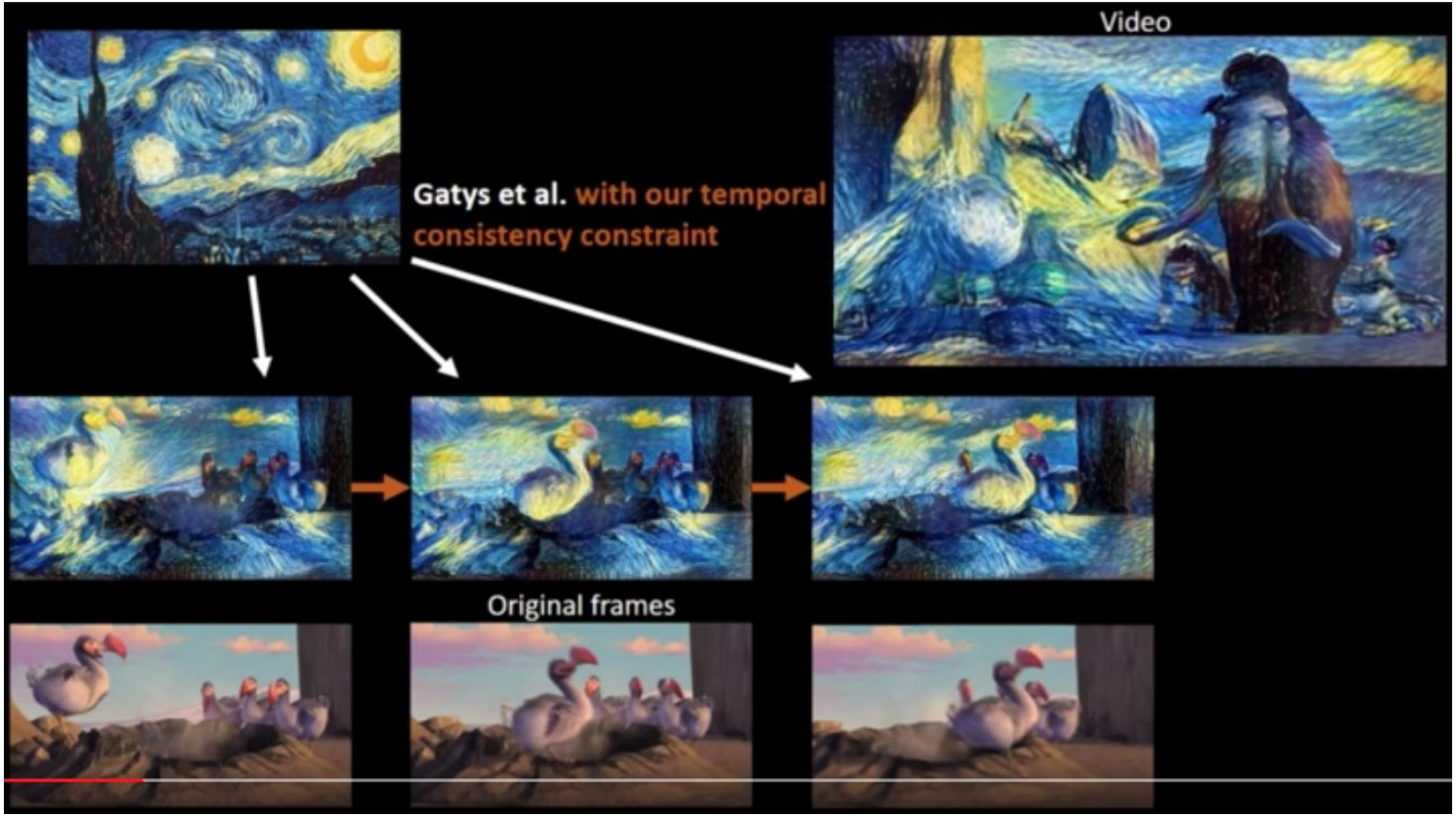
# Snagging Parking Spaces with Mask R-CNN and Python



https://medium.com/@ageitgey/snagging-parking-spaces-with-mask-r-cnn-and-python-955f2231c400

GROUPON

# Artistic style transfer for videos




https://www.youtube.com/watch?v=Khuj4ASldmU

GROUPON

# A Style-Based Generator Architecture for Generative Adversarial Networks


https://www.youtube.com/watch?v=kSLJriaOumA

GROUPON

# AI Reconstructs Photos with Realistic Results


https://www.youtube.com/watch?v=gg0F5JjKmhA

GROUPON

# Making Money on Neural Network DCGAN

## AI portrait sold for .5M USD

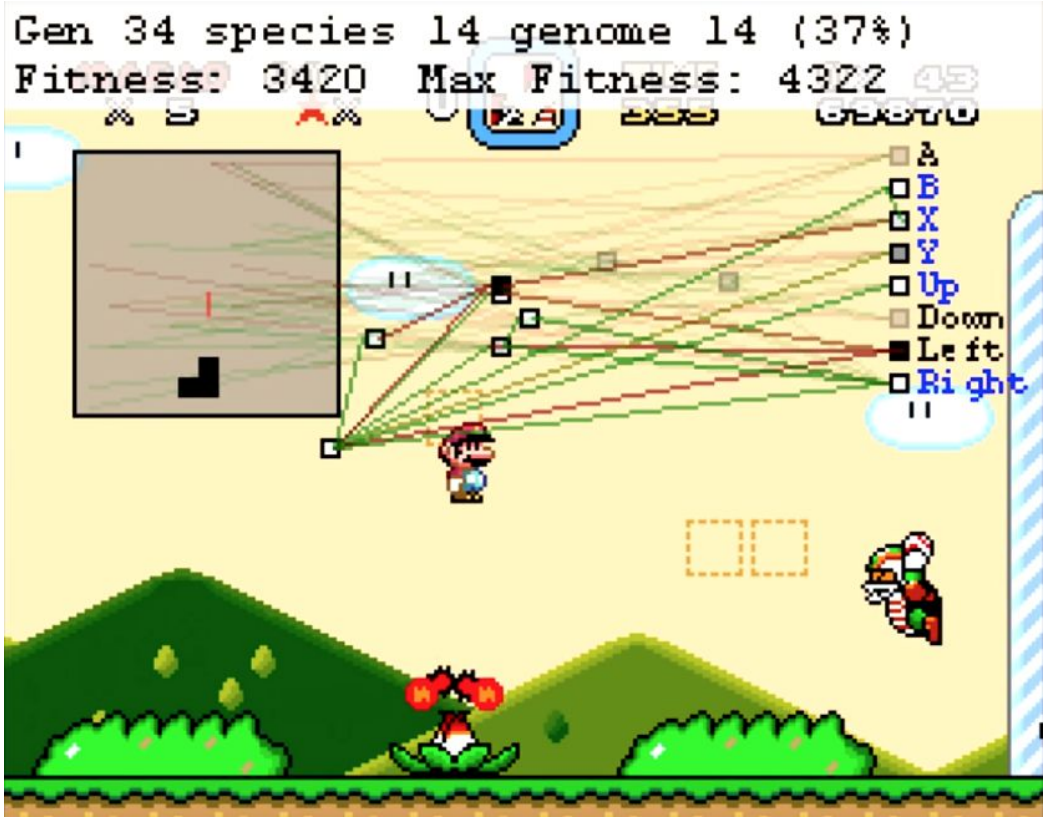
https://github.com/robbiebarrat/art-DCGAN
https://twitter.com/drbeef_/status/1055285640420483073
https://www.forbes.com/sites/williamfalcon/2018/10/25/what-happens-now-that-an-ai-generated-painting-sold-for-432500

GROUPON

# GAN in GROUPON


https://comixify.ii.pw.edu.pl

# Neural Networks in Games



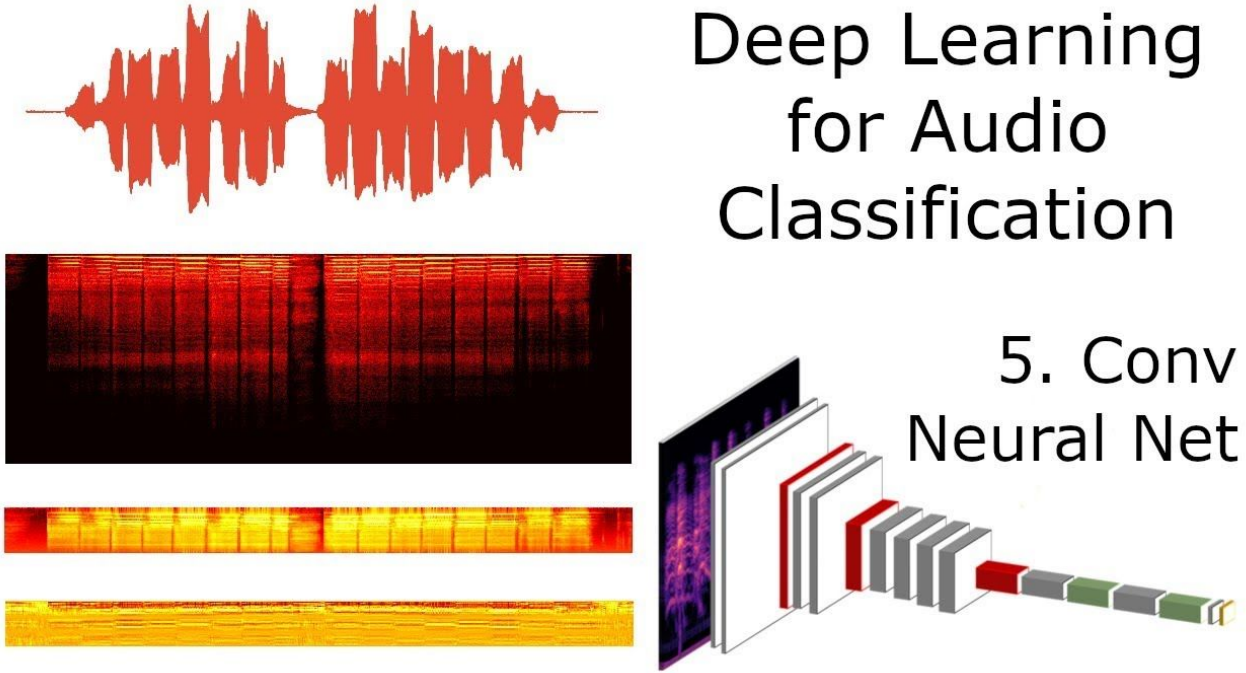

MarI/O - Machine Learning for Video Games
https://www.youtube.com/watch?v=qv6UVOQ0F44

## NEAT

## Neuroevolution of Augmenting Topologies

GROUPON

# Deep Learning for Audio Classification




https://www.youtube.com/playlist?list=PLhA3b2k8R3t2Ng1WW_7MiXeh1pfQJQi_P

GROUPON

# Deep Learning SR + TTS


**Google Duplex: A.I. Assistant Calls Local Businesses To Make Appointments**

https://www.youtube.com/watch?v=D5VN56jQMWM&t=48s

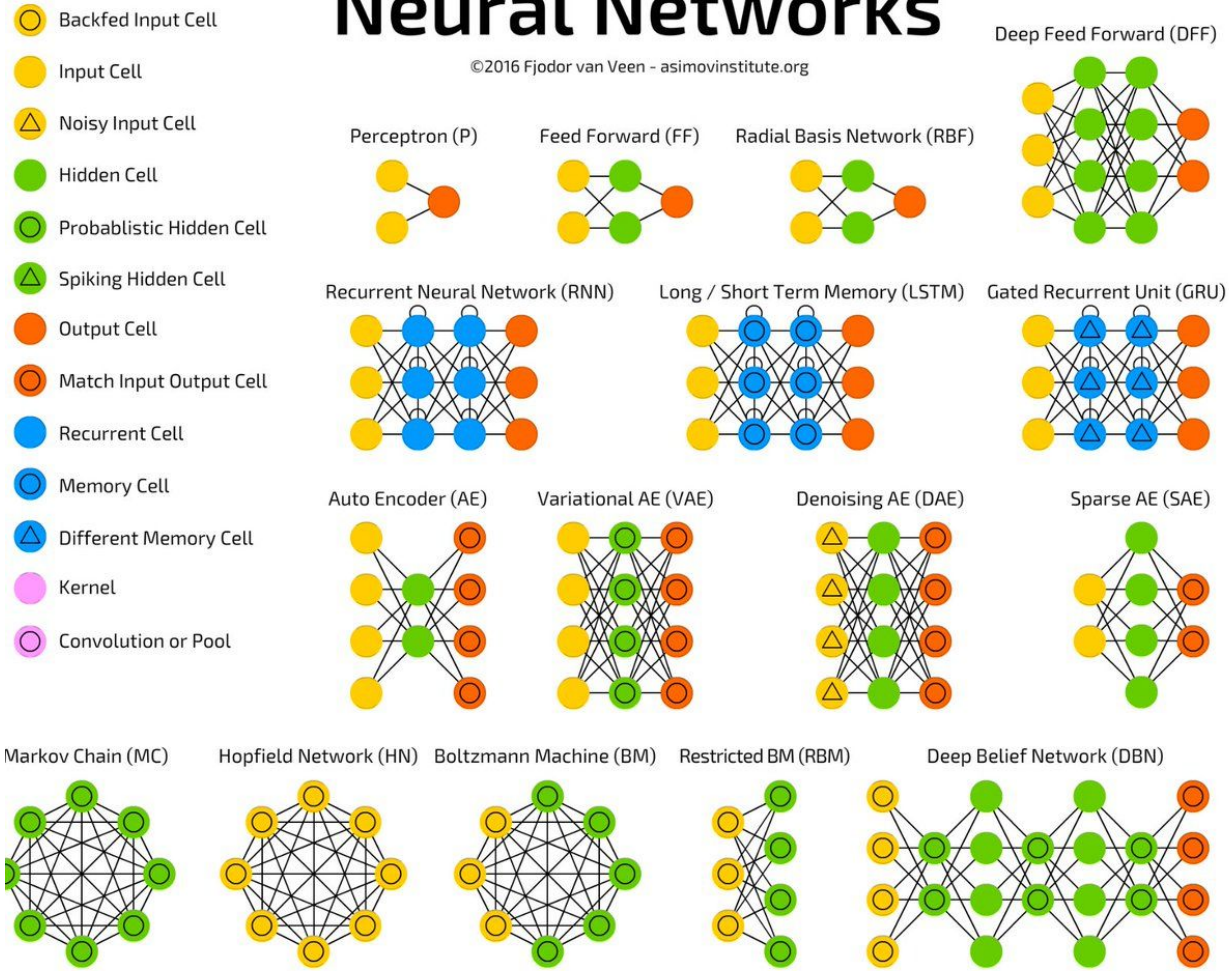
GROUPON

*A mostly complete chart of*

# Neural Networks

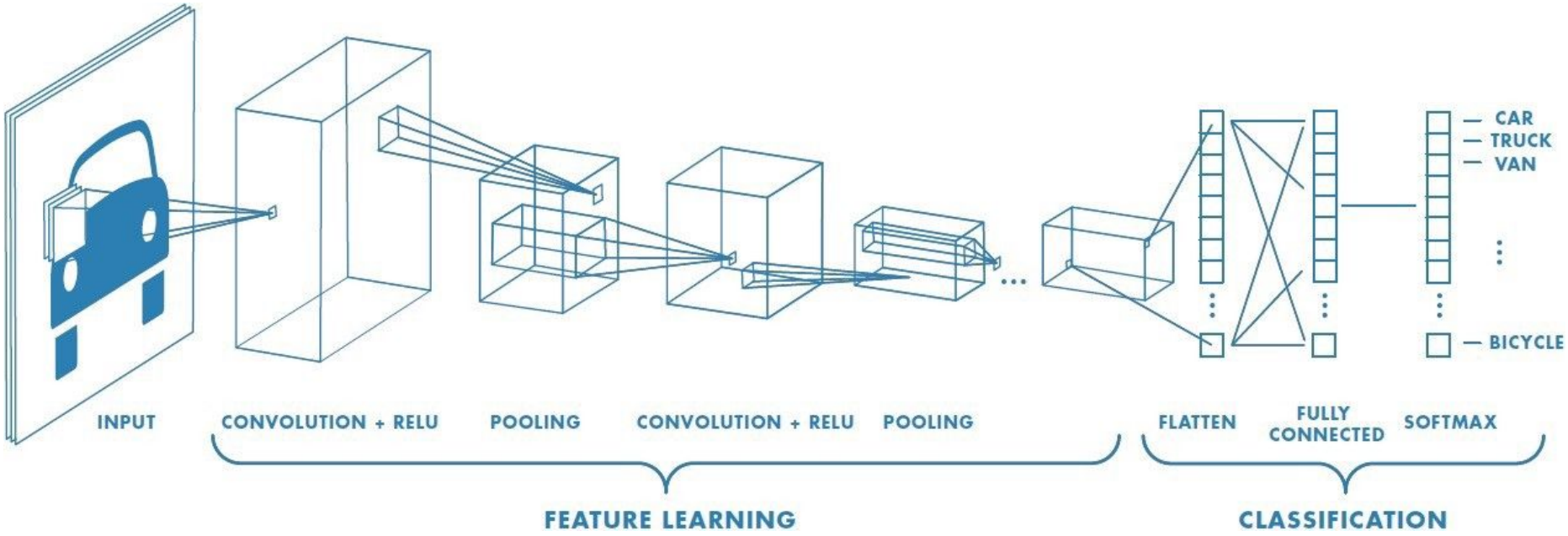


- Backfed Input Cell
- Input Cell
- Noisy Input Cell
- Hidden Cell
- Probablistic Hidden Cell
- Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- Different Memory Cell
- Kernel
- Convolution or Pool




GROUPON

# Topology Convolutional Network




GROUPON

# Keras Model



```
# Define RNN - Recursive Neural Network
def RNN():
    inputs = Input(name = 'inputs', shape = [max_len])
    layer = Embedding(max_words, 50, input_length = max_len)(inputs)
    layer = LSTM(64)(layer)
    layer = Dense(256, name = 'FC1')(layer)
    layer = Activation('relu')(layer)
    layer = Dropout(0.5)(layer)
    layer = Dense(1, name = 'out_layer')(layer)
    layer = Activation('sigmoid')(layer)
    model = Model(inputs = inputs, outputs = layer)
    return model
```

# Label Encoder and One Hot Encoder

```
le = LabelEncoder()              # Create Label Encoder
Y = le.fit_transform(df.Label) # Set Labels (ham, spam) -> (0, 1)
```

## Label Encoding

| Food Name | Categorical # | Calories |
|---|---|---|
| Apple | 1 | 95 |
| Chicken | 2 | 231 |
| Broccoli | 3 | 50 |

→

## One Hot Encoding

| Apple | Chicken | Broccoli | Calories |
|---|---|---|---|
| 1 | 0 | 0 | 95 |
| 0 | 1 | 0 | 231 |
| 0 | 0 | 1 | 50 |

GROUPON

# Tokenization

Natural Language
Analyses with NLTK

```
>>> import nltk
>>> sentence = """At eight o'clock on Thursday morning
... Arthur didn't feel very good."""
>>> tokens = nltk.word_tokenize(sentence)
>>> tokens
['At', 'eight', "o'clock", 'on', 'Thursday', 'morning',
'Arthur', 'did', "n't", 'feel', 'very', 'good', '.']
```

GROUPON

# Tokenization in code ...

```
# Tokenize the data and convert the text to sequences
# Add padding to ensure that all the sequences have the same shape
# There are many ways of taking the max_len and here an arbitrary length of 150 is chosen
max_words = 1000
max_len = 150

# Split Sentences into Tokens
tok = Tokenizer(num_words = max_words)
tok.fit_on_texts(X_train)

# Transform sequence of senteces in sequence of integers
sequences = tok.texts_to_sequences(X_train)
# Create Input to Neural Network
sequences_matrix = sequence.pad_sequences(sequences, maxlen = max_len)
```

```
# Raw Textual Data
print(X_train[1])
# After Tokenization
print(sequences[1])
# Matrix with Tokens (Proper Input to Neural Network)
print(sequences_matrix[1])
```

```
Ok lar... Joking wif u oni...
[1, 60, 8, 16, 3, 83]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  1 60  8 16  3 83]
```

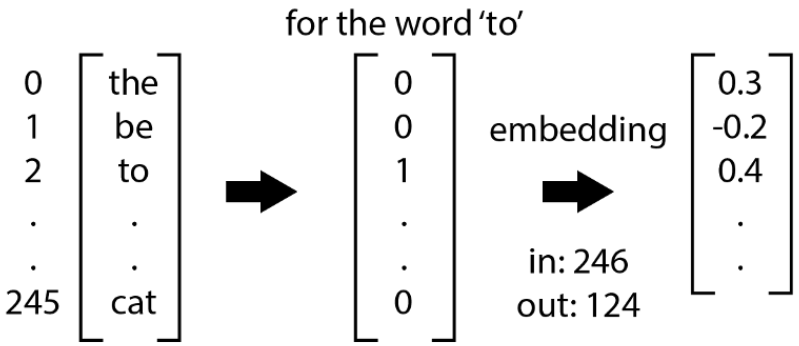
GROUPON

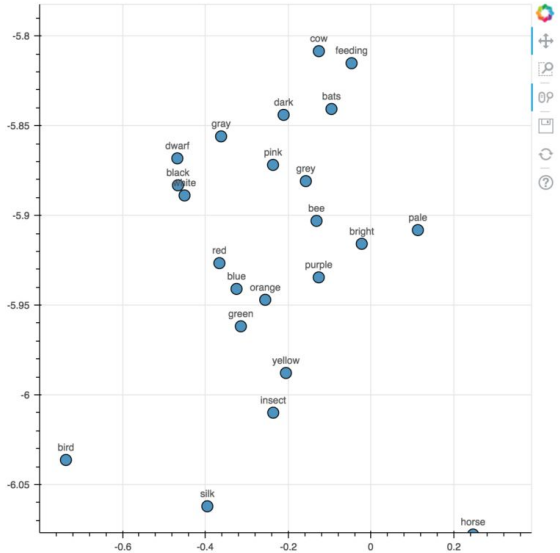
# Keras Model

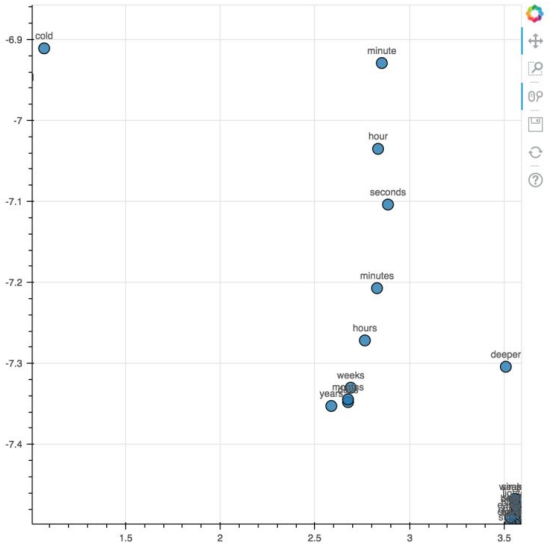


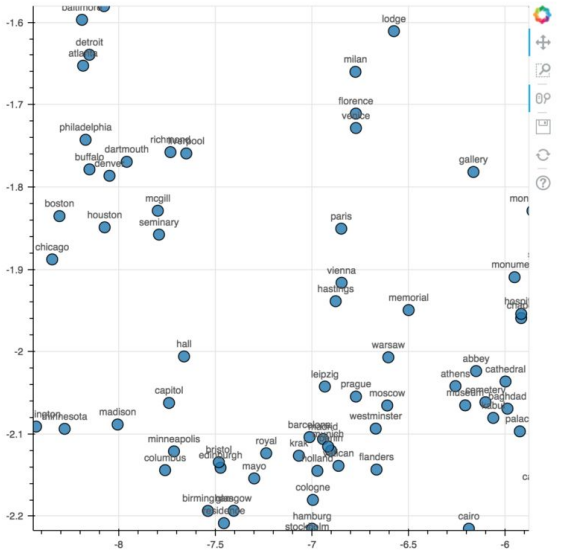
```
# Define RNN - Recursive Neural Network
def RNN():
    inputs = Input(name = 'inputs', shape = [max_len])
    layer = Embedding(max_words, 50, input_length = max_len)(inputs)
    layer = LSTM(64)(layer)
    layer = Dense(256, name = 'FC1')(layer)
    layer = Activation('relu')(layer)
    layer = Dropout(0.5)(layer)
    layer = Dense(1, name = 'out_layer')(layer)
    layer = Activation('sigmoid')(layer)
    model = Model(inputs = inputs, outputs = layer)
    return model
```

# Word Embeddings e.g. Word2Vec



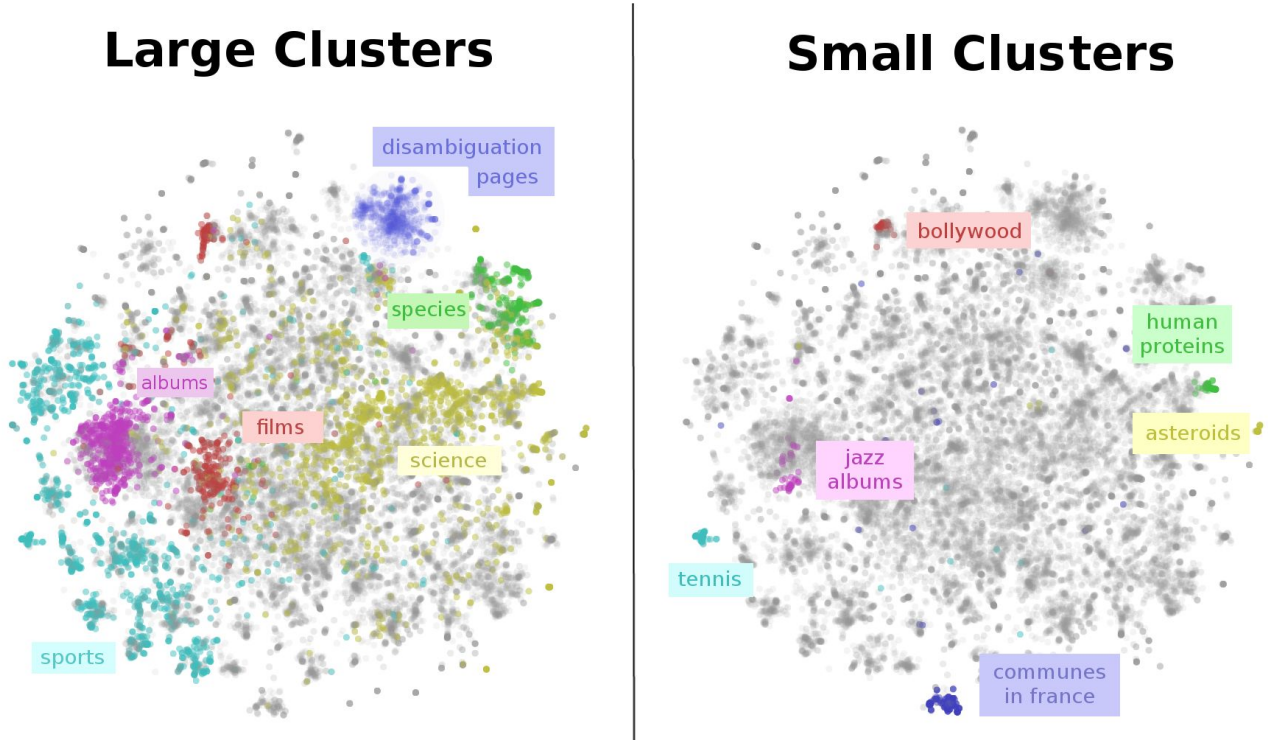




https://nlpforhackers.io/word-embeddings

GROUPON

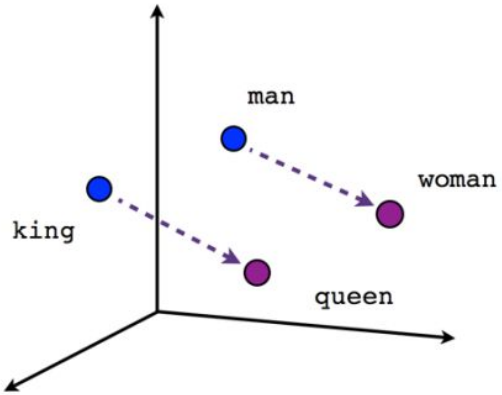
# Map of Wikipedia (t-SNE)



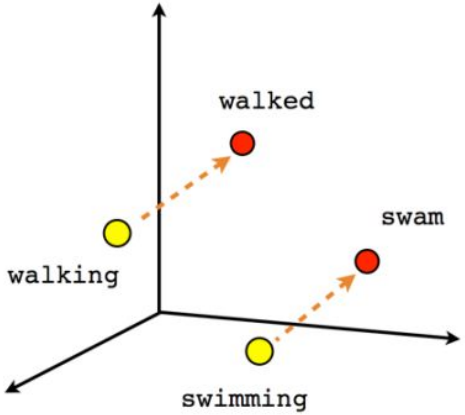

http://colah.github.io/posts/2015-01-Visualizing-Representations

GROUPON

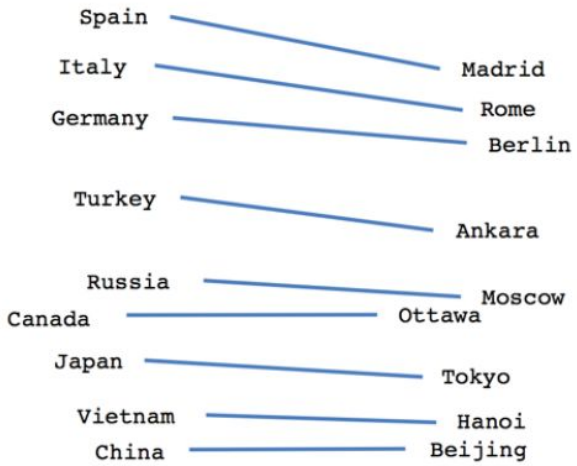
# Word2Vec - Hidden Semantic Relationships




Male-Female




Verb tense




Country-Capital

https://www.tensorflow.org/tutorials/representation/word2vec

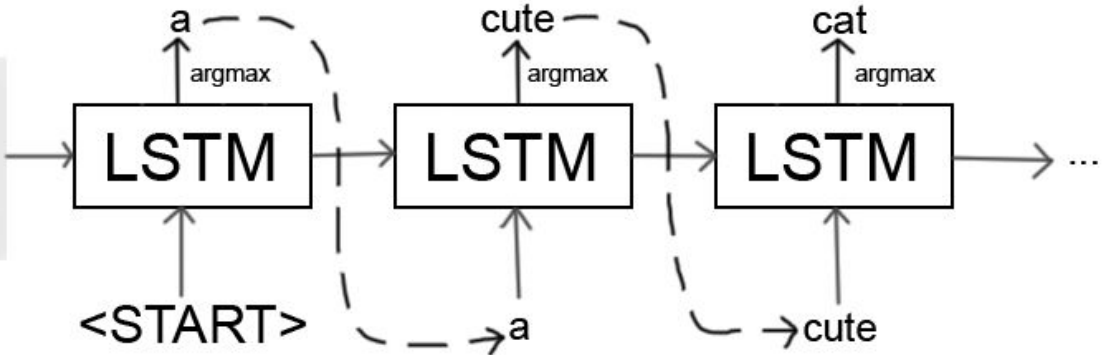
GROUPON

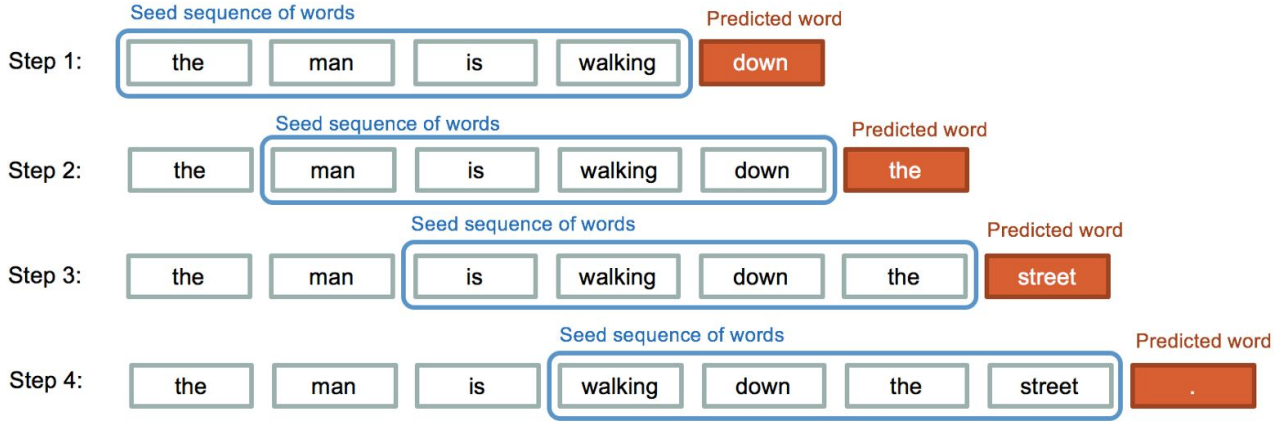
# Keras Model



```
# Define RNN - Recursive Neural Network
def RNN():
    inputs = Input(name = 'inputs', shape = [max_len])
    layer = Embedding(max_words, 50, input_length = max_len)(inputs)
    layer = LSTM(64)(layer)
    layer = Dense(256, name = 'FC1')(layer)
    layer = Activation('relu')(layer)
    layer = Dropout(0.5)(layer)
    layer = Dense(1, name = 'out_layer')(layer)
    layer = Activation('sigmoid')(layer)
    model = Model(inputs = inputs, outputs = layer)
    return model
```

# LSTM - Long Short Term Memory




https://medium.com/@david.campion/text-generation-using-bidirectional-lstm-and-doc2vec-models-1-3-8979eb65cb3a
https://luckytoilet.wordpress.com/tag/lstm

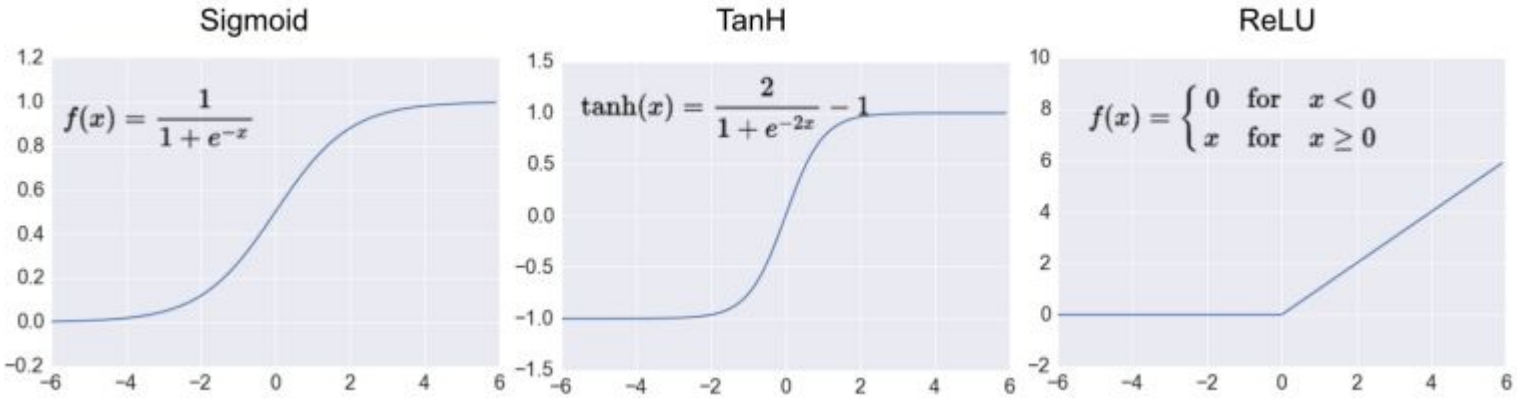
GROUPON

# Keras Model



```
# Define RNN - Recursive Neural Network
def RNN():
    inputs = Input(name = 'inputs', shape = [max_len])
    layer = Embedding(max_words, 50, input_length = max_len)(inputs)
    layer = LSTM(64)(layer)
    layer = Dense(256, name = 'FC1')(layer)
    layer = Activation('relu')(layer)
    layer = Dropout(0.5)(layer)
    layer = Dense(1, name = 'out_layer')(layer)
    layer = Activation('sigmoid')(layer)
    model = Model(inputs = inputs, outputs = layer)
    return model
```

# Activation Functions




https://www.kdnuggets.com/2017/09/neural-network-foundations-explained-activation-function.html
https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

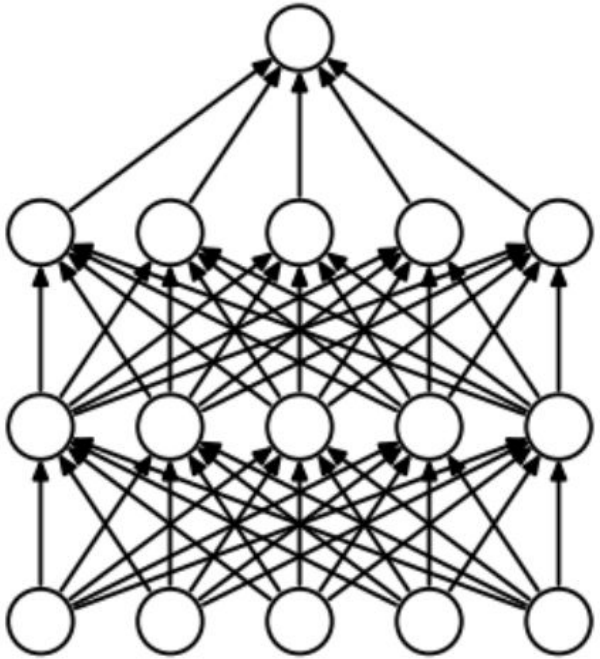
GROUPON

# Keras Model

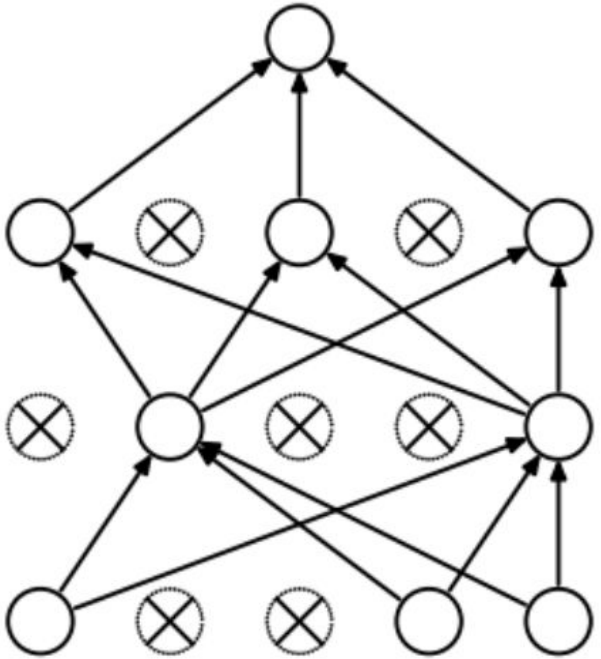


```
# Define RNN - Recursive Neural Network
def RNN():
    inputs = Input(name = 'inputs', shape = [max_len])
    layer = Embedding(max_words, 50, input_length = max_len)(inputs)
    layer = LSTM(64)(layer)
    layer = Dense(256, name = 'FC1')(layer)
    layer = Activation('relu')(layer)
    layer = Dropout(0.5)(layer)
    layer = Dense(1, name = 'out_layer')(layer)
    layer = Activation('sigmoid')(layer)
    model = Model(inputs = inputs, outputs = layer)
    return model
```

# Fully Connected vs Drop Out



(a) Standard Neural Net



(b) After applying dropout.

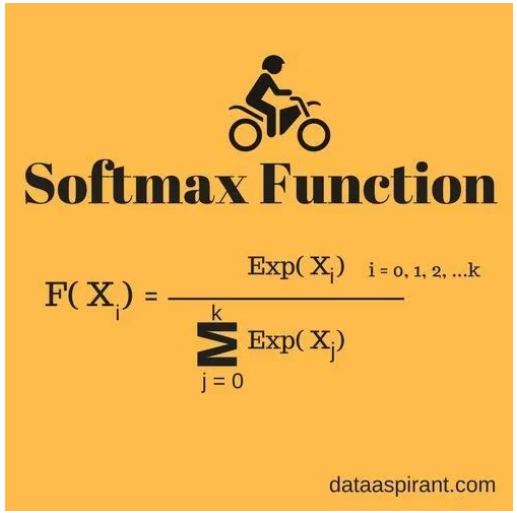
GROUPON

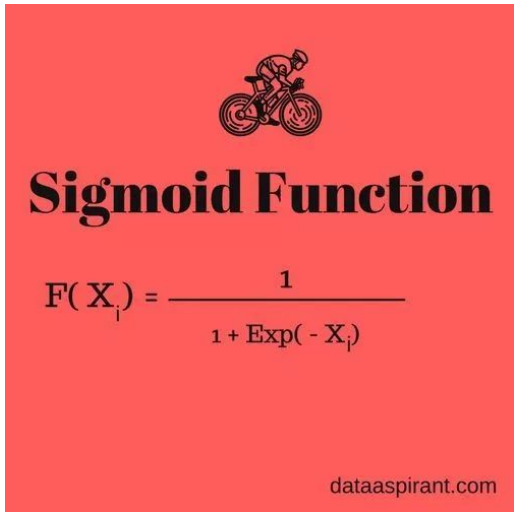
# Keras Model



```
# Define RNN - Recursive Neural Network
def RNN():
    inputs = Input(name = 'inputs', shape = [max_len])
    layer = Embedding(max_words, 50, input_length = max_len)(inputs)
    layer = LSTM(64)(layer)
    layer = Dense(256, name = 'FC1')(layer)
    layer = Activation('relu')(layer)
    layer = Dropout(0.5)(layer)
    layer = Dense(1, name = 'out_layer')(layer)
    layer = Activation('sigmoid')(layer)
    model = Model(inputs = inputs, outputs = layer)
    return model
```

# Softmax vs Sigmoid



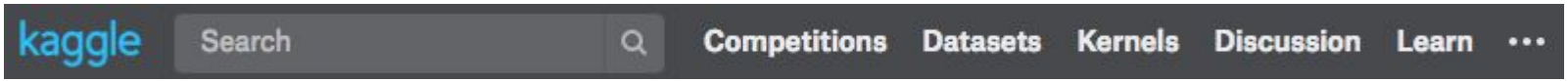




| | Softmax Function | Sigmoid Function |
|---|---|---|
| 1 | Used for multi-classification in logistic regression model. | Used for binary classification in logistic regression model. |
| 2 | The probabilities sum will be 1 | The probabilities sum need not be 1. |
| 3 | Used in the different layers of neural networks. | Used as activation function while building neural networks. |
| 4 | The high value will have the higher probability than other values. | The high value will have the high probability but not the higher probability. |

http://dataaspirant.com/2017/03/07/difference-between-softmax-function-and-sigmoid-function GROUPON

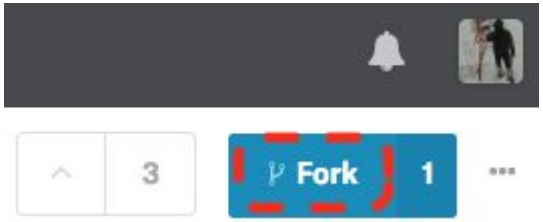
# Kaggle

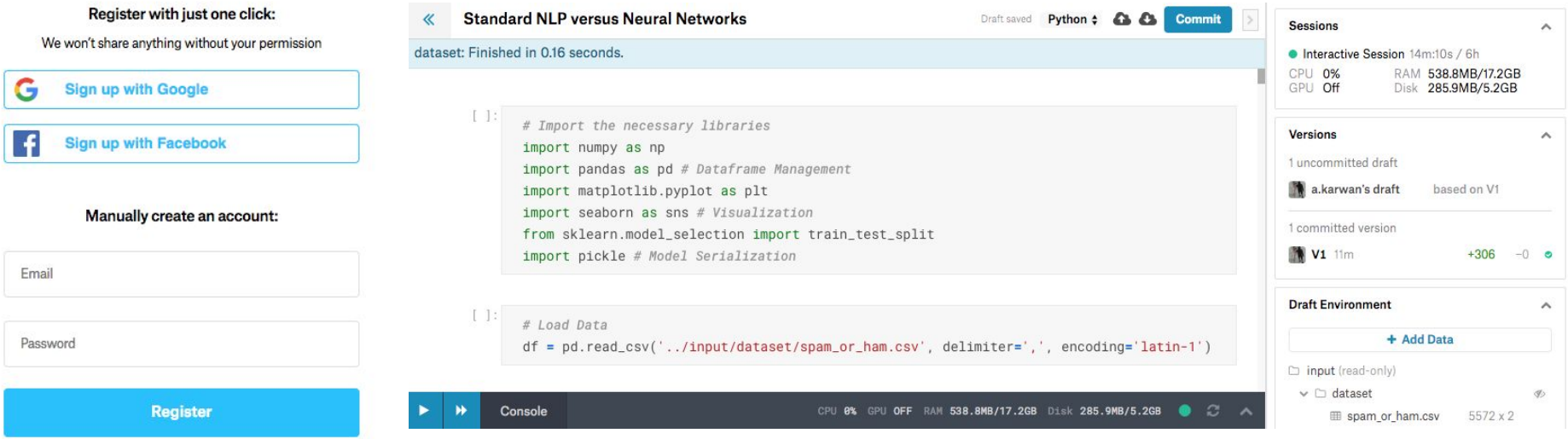
- ★ DataSets
- ★ Contests
- ★ Self-Learning
- ★ Recruitment
- ★ Algorithms for Companies


GROUPON

# Kaggle (codes)



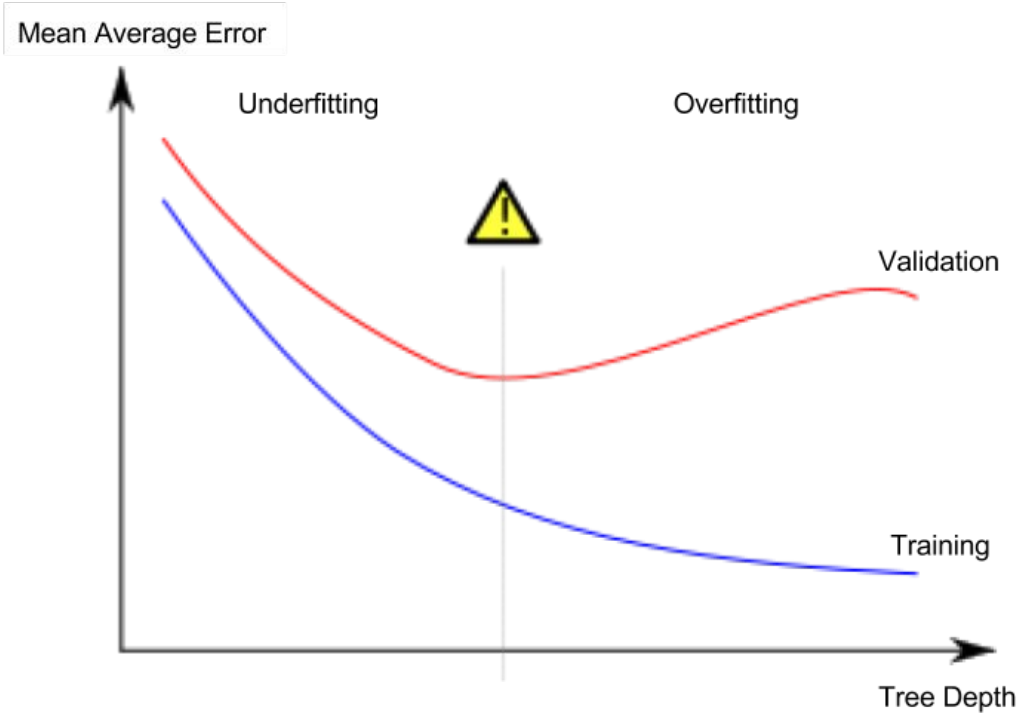



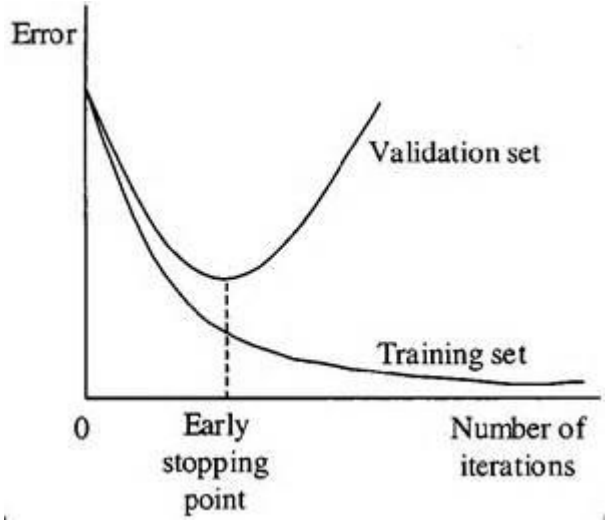


https://www.kaggle.com/akarwan/standard-nlp-versus-neural-networks

GROUPON

# Train Neural Network

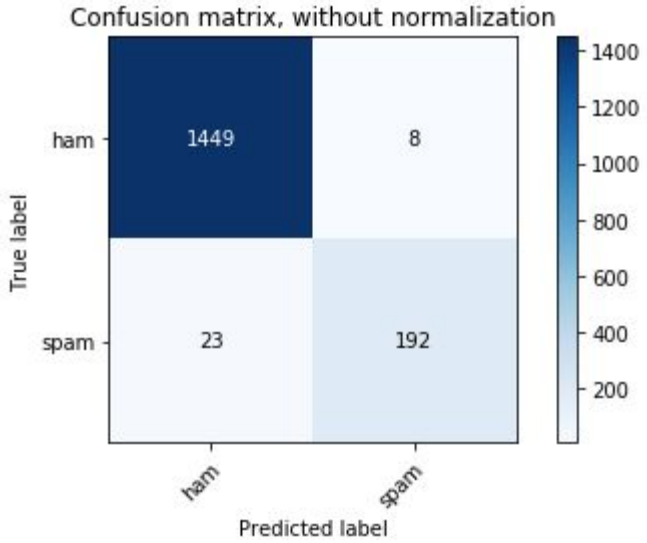



## Avoid Overfit

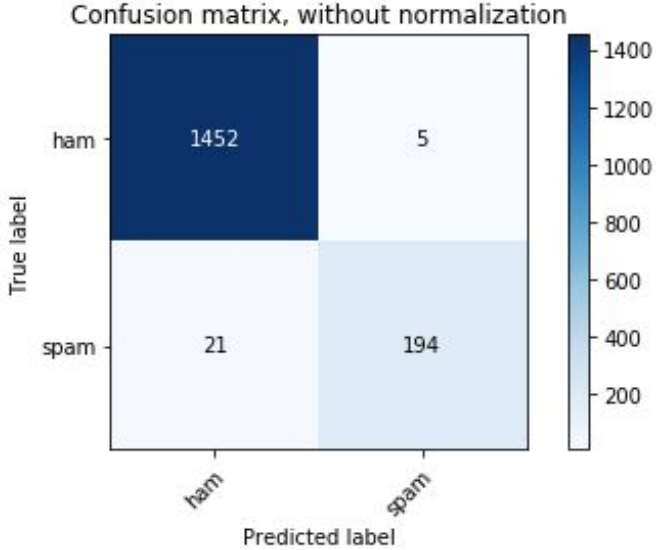
GROUPON

# Deep Learning vs Standard NLP




## RNN

```
4s Epoch 3.
Accuracy: 0.980
```

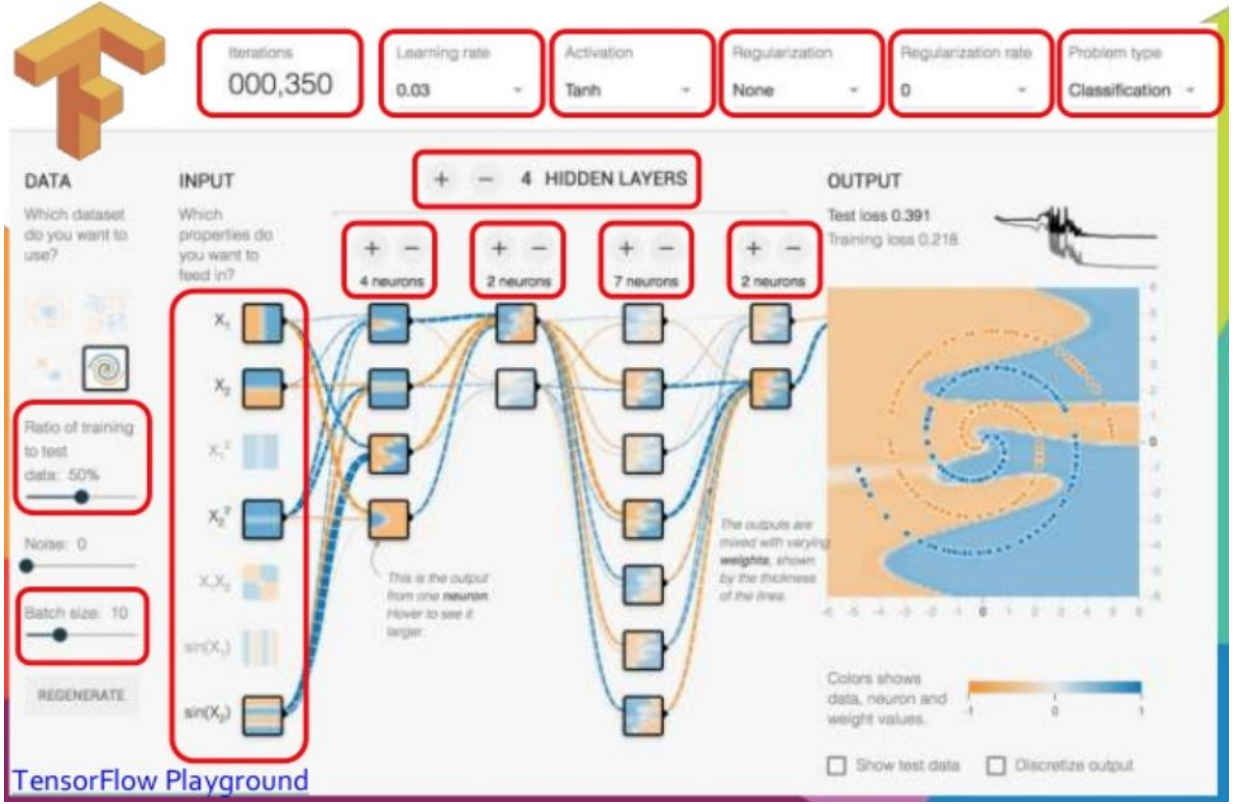



## SVM

```
Training done in 5.247s
Model final score: 0.984
```

GROUPON

# Red Box = Hyperparameter




https://playground.tensorflow.org

https://www.slideshare.net/SigOpt/tips-and-techniques-for-hyperparameter-optimization

GROUPON

# Part 3

## Real Life Examples

# MORE CATEGORIES

Categorize **consumer financial complaints** into **11 classes**

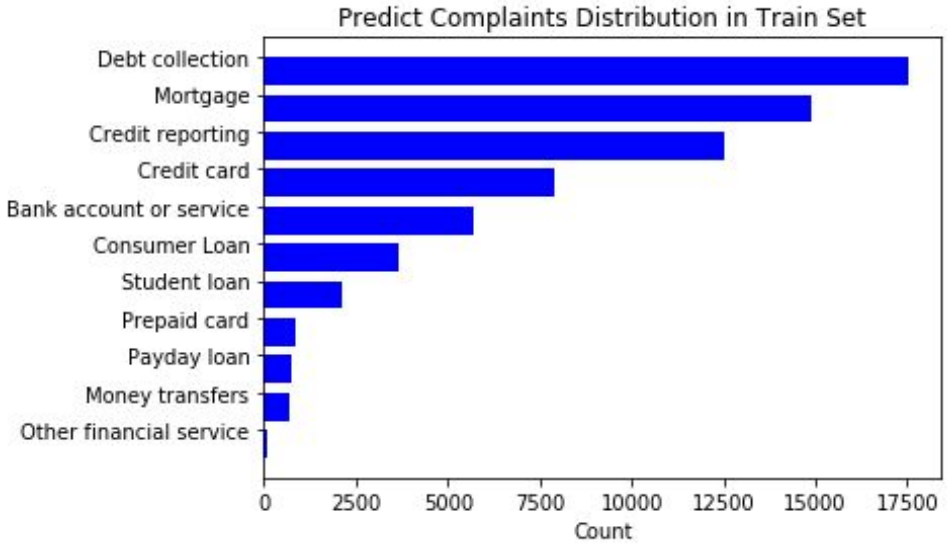

**Randomly ~9% Acc**

Thousands of the consumer complaints weekly about financial products and services

Need classification before sending final respond within 15 days

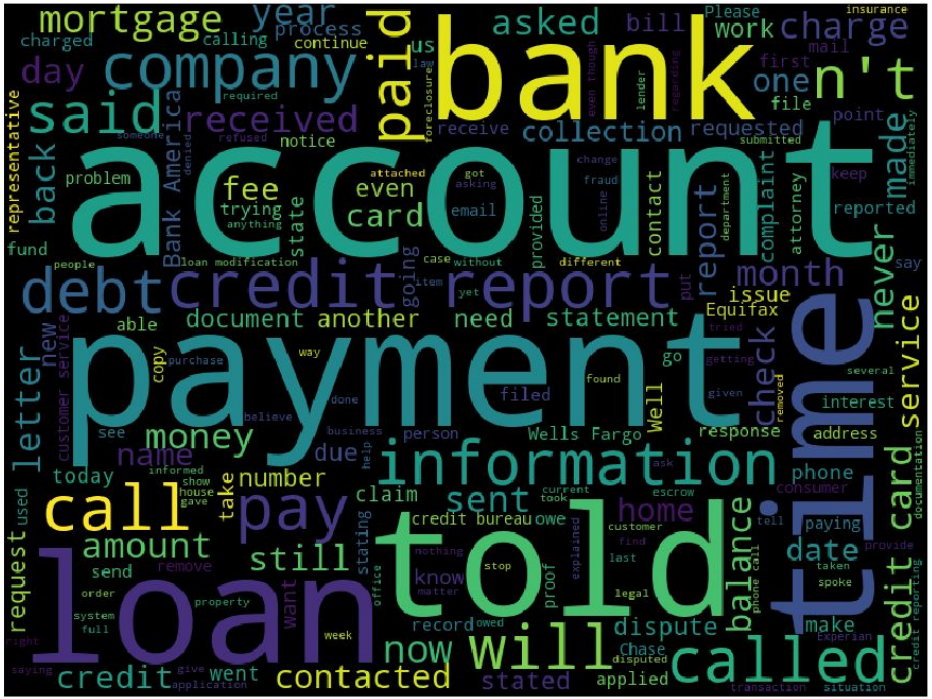
GROUPON

# Data Train / Test 66.8k / 1k





GROUPON



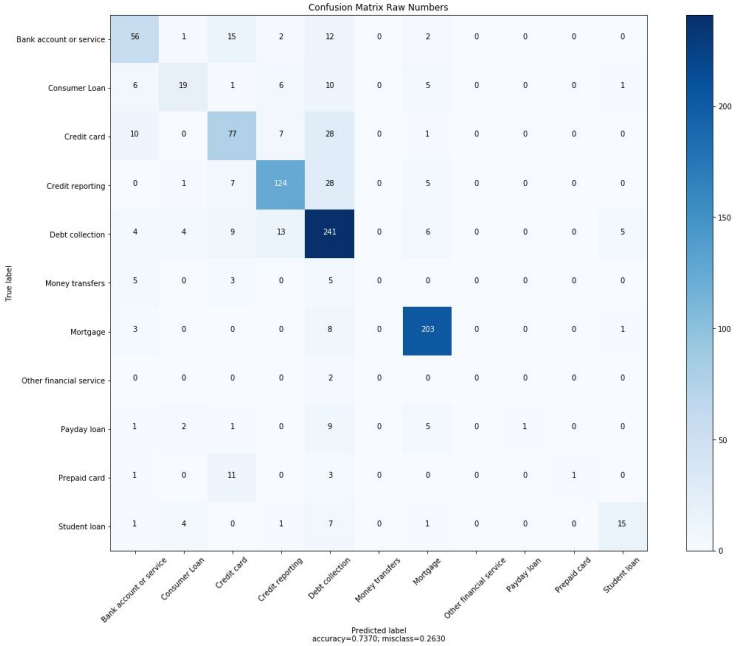

# Test 1k Data

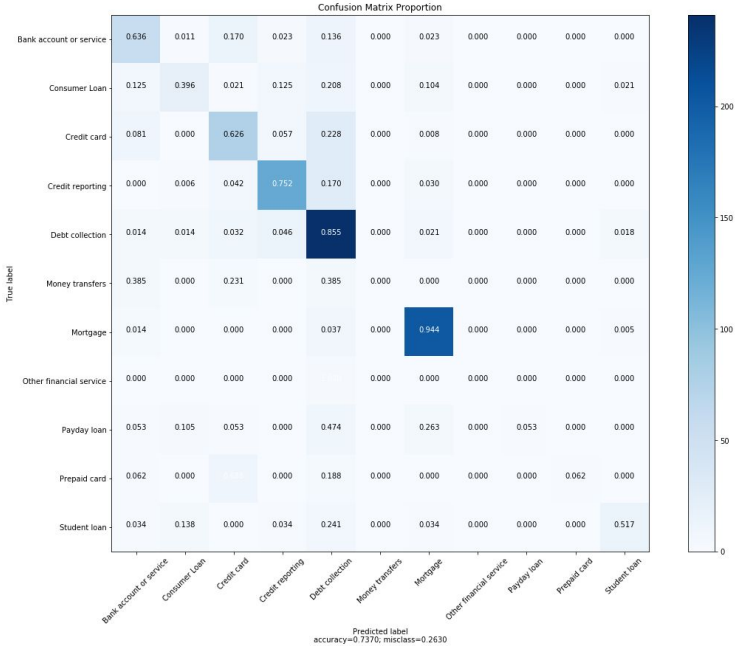
```
'Debt collection'
```

```
'This creditor is harrasing me at work. They are calling my children repeatedly. \n'
```

GROUPON

# TensorFlow Model, train in 2 minutes




# Final Prediction 73.7% on 1000 samples

GROUPON

RAW DATA | CLEANING | FEATURE EXTRACTION | ML MODELS | TUNING | **API**

Transformation
Data Filtering
Removing:
- tags
- duplicates
Formatting

Stemming
Case Folding
Removing:
- Stop words
- Double Letters

Add New Features

TF-IDF
Word2Vec

Data Enrichment
(non Textual)

Other embeddings
FastText
ELMo

**Kerras**
**BRNN & LSTM**

Naive Bayes
Regression
Linear SVM
SVM (Kernel)
Random Forests
etc.

DL Topology

Hyperparameters
Grid Search
**Ensemble methods**

Confusion Matrix

Final Model
Ready to
Deploy

Retrain and
Update Model

# e.g. standard NLP approach

```
model = Pipeline([('vect', CountVectorizer(ngram_range=(1,3))),
                  ('tfidf', TfidfTransformer(use_idf=False)),
                  ('clf', CalibratedClassifierCV(LinearSVC(), cv=2))])
```

★ Category label
★ Probability of the correct categorization

GROUPON

# Deep Learning, model in Keras

Keras

```
# Topology
inp = Input(shape=(maxlen, )) # maxlen=300
embed_size = 128
x = Embedding(max_features, embed_size)(inp)
x = Bidirectional(LSTM(40, return_sequences=True, name='lstm_layer'))(x)
x = GlobalMaxPool1D()(x)
x = Dropout(0.1)(x)
x = Dense(20, activation="tanh")(x) # 50
x = Dropout(0.1)(x)
x = Dense(11, activation="softmax")(x) # original "sigmoid"

# Prepare Model
model = Model(inputs=inp, outputs=x)
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

batch_size = 32 # original 32
epochs = 2
```

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_7 (InputLayer) | (None, 300) | 0 |
| embedding_6 (Embedding) | (None, 300, 128) | 2560000 |
| bidirectional_6 (Bidirection | (None, 300, 80) | 54080 |
| global_max_pooling1d_6 (Glob | (None, 80) | 0 |
| dropout_11 (Dropout) | (None, 80) | 0 |
| dense_11 (Dense) | (None, 20) | 1620 |
| dropout_12 (Dropout) | (None, 20) | 0 |
| dense_12 (Dense) | (None, 11) | 231 |

Total params: 2,615,931
Trainable params: 2,615,931
Non-trainable params: 0

## training ~15 minutes

playground.tensorflow.org Run and Modify Neural Networks, no Coding Skills Required

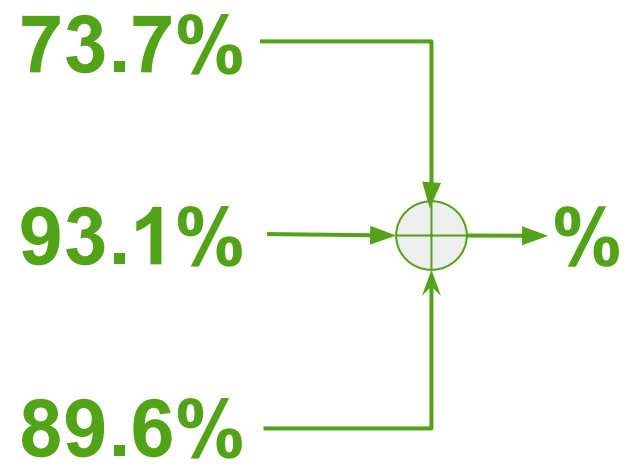
GROUPON

# Github vs Standard vs Keras

GitHub multi-class-text-classification-cnn

```
model = Pipeline([('vect', CountVectorizer(ngram_range=(1,3))),
                  ('tfidf', TfidfTransformer(use_idf=False)),
                  ('clf', CalibratedClassifierCV(LinearSVC(), cv=2))])
```

Keras **RNN with BLSTM**

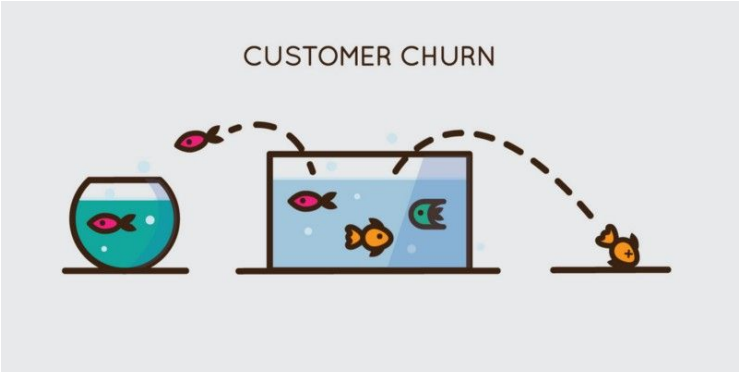



- ★ Ensemble method
- ★ Imbalanced data (Imputation, e.g. SMOTE Algorithm)
- ★ Provide correct annotations

GROUPON

# Churn Modeling

$$\frac{\text{USERS LOST THIS MONTH}}{(\text{USERS AT START OF MONTH} + \text{USERS ADDED THIS MONTH})} = \text{MONTHLY CHURN RATE}$$





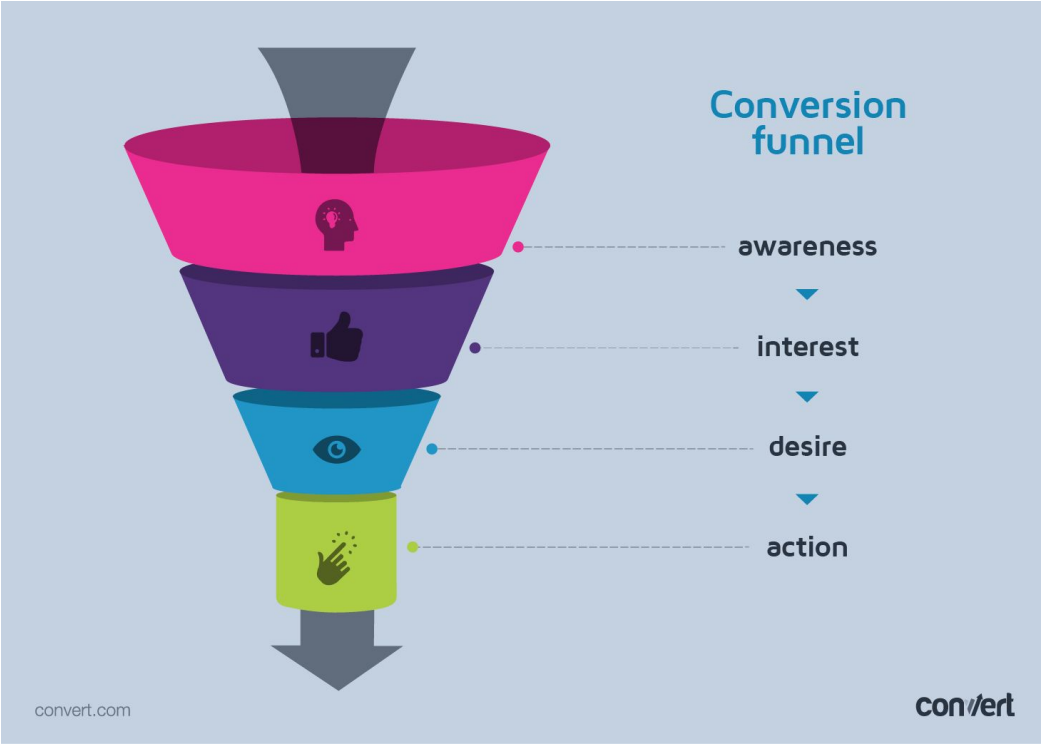
GROUPON

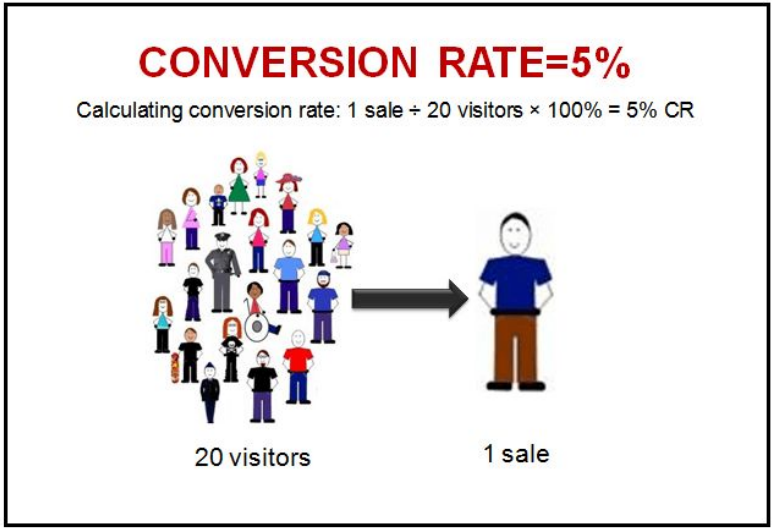
# SNA - Social Network Analysis

- ★ Leaders
- ★ Bridges

GROUPON

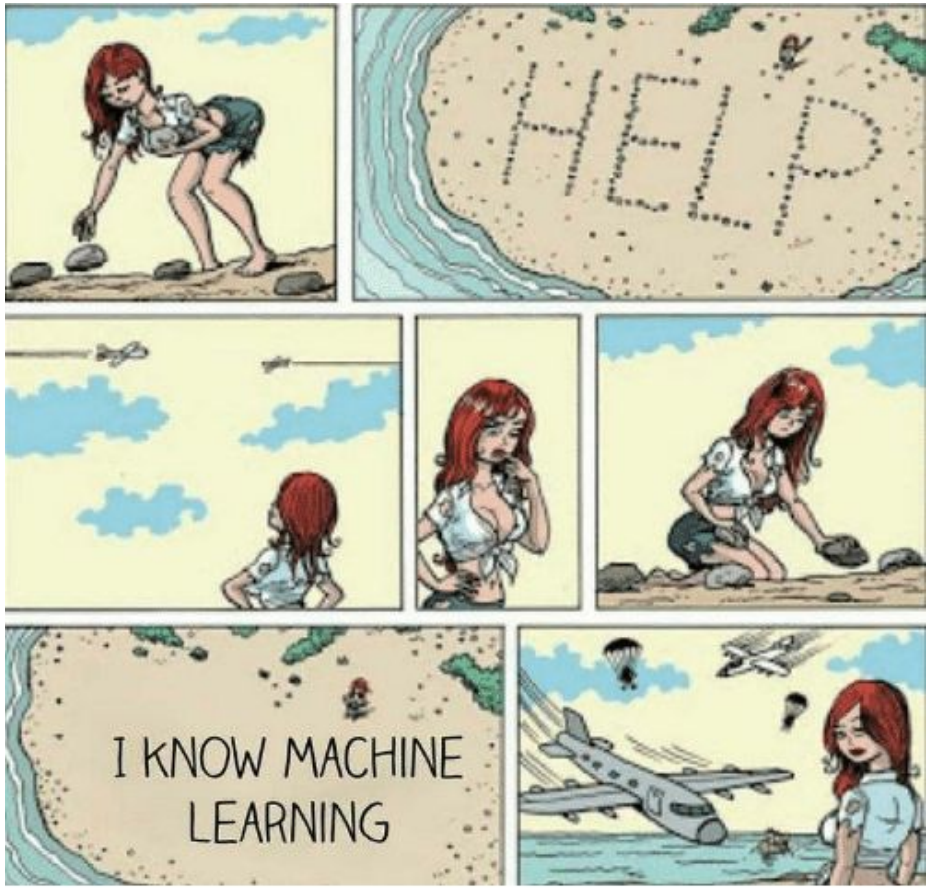
# Conversion Funnel Analysis




https://www.wakefly.com/blog/whats-important-paid-search-click-through-rate-or-conversion-rate
https://beloandco.com/the-new-top-paths-to-conversion-report

GROUPON




Machine Learning is the key

VERIFIED CERTIFICATE OF PARTICIPATION

Katowice, March 12 2019

GROUPON

Has successfully completed the

AUTOMATIC DOCUMENT CATEGORIZATION WITH MACHINE LEARNING TECHNIQUES IN PYTHON WORKSHOP

**Trainers:**

Adam Karwan

Aleksander Zajchowski

Roksana Tomanek

Sviatoslav Somov

GSO Technology, Robotics & Automation