

AI WORKSHOP

How we're using Data Science at Groupon

Adam, Aleksander, Roksana and Sviatoslav

Katowice, 12.03.2019

GROUPON[®]

Part 1

Standard Machine Learning

Machine Learning



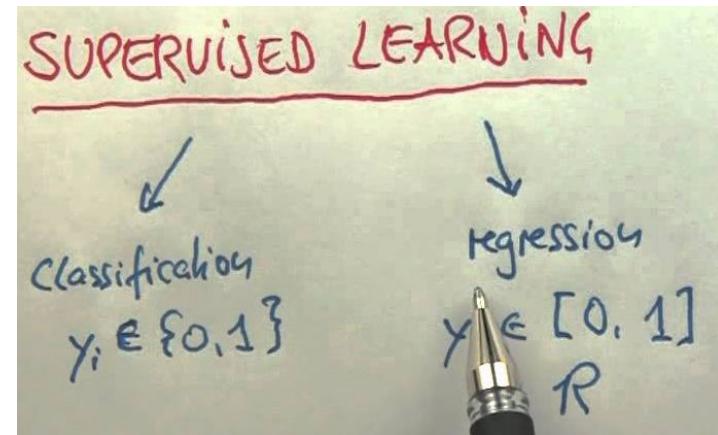
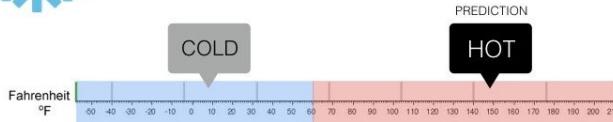
Regression

What is the temperature going to be tomorrow?

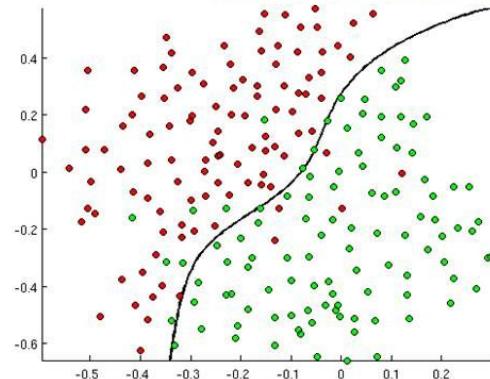


Classification

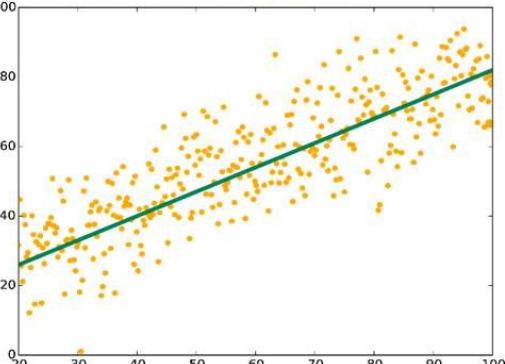
Will it be Cold or Hot tomorrow?



What is the Difference Between



Classification



Regression

Run Machine Learning Models

Laptop



Cloud

kaggle.com



GROUPON®

Kaggle

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

Kaggle is the place to do data science projects

See how it works 



kaggle

Search



Competitions

Datasets

Kernels

Discussion

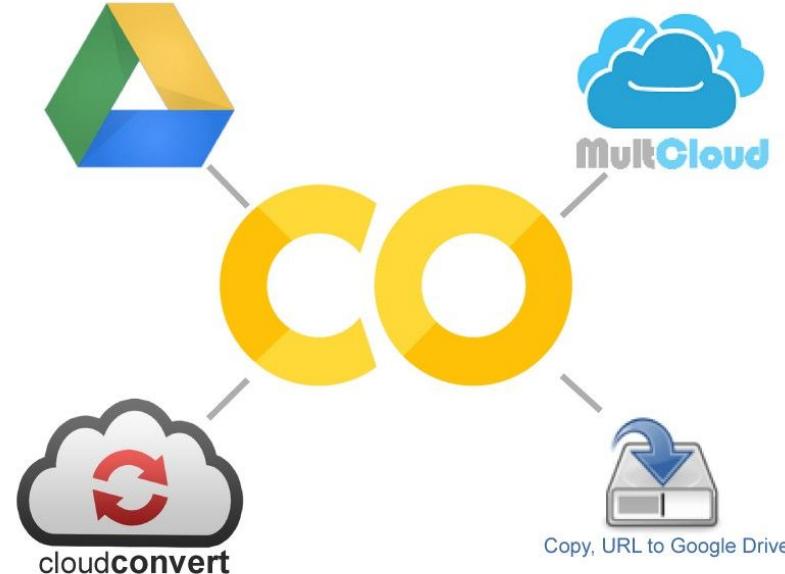
Learn

...

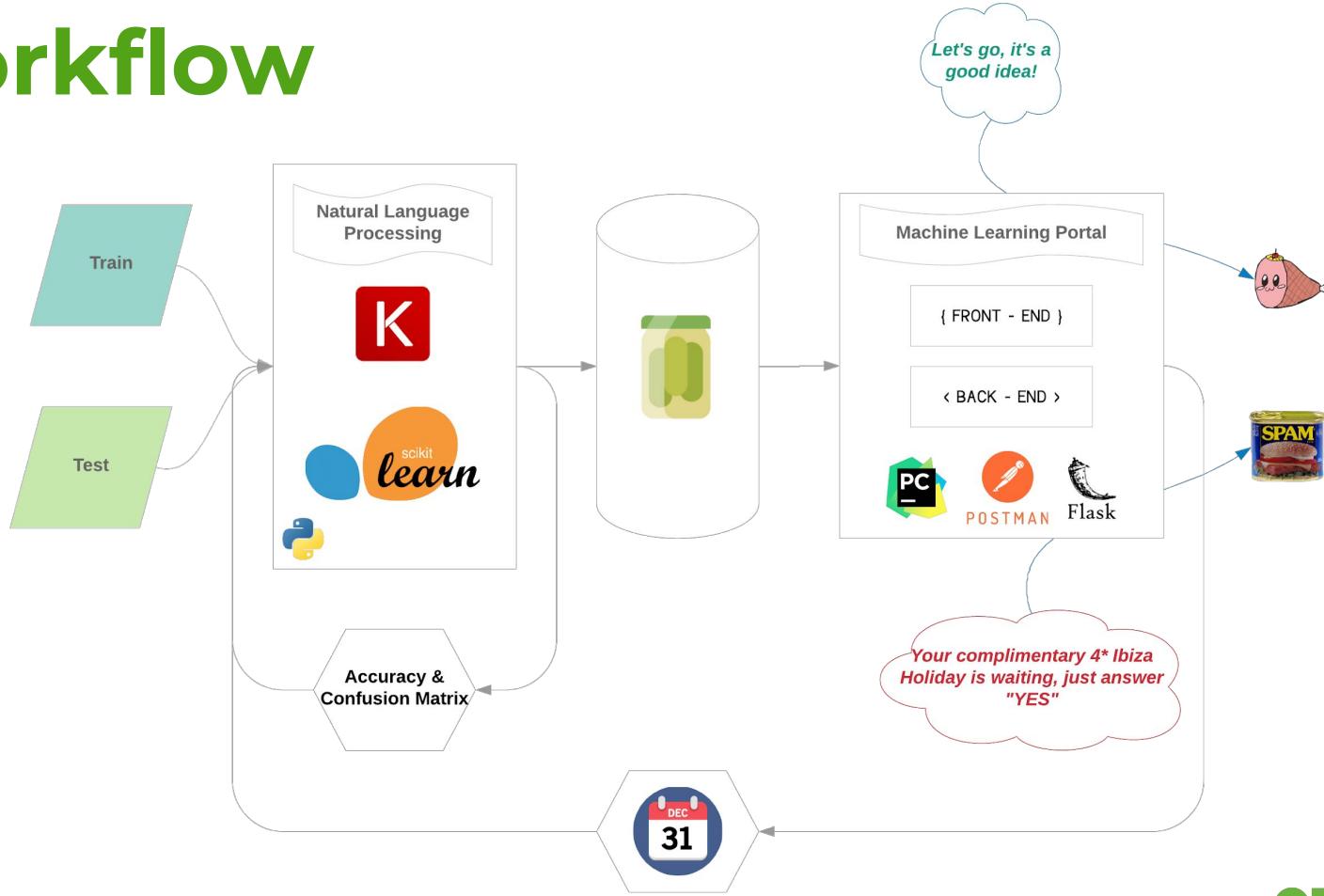
GROUPON®

colab.research.google.com

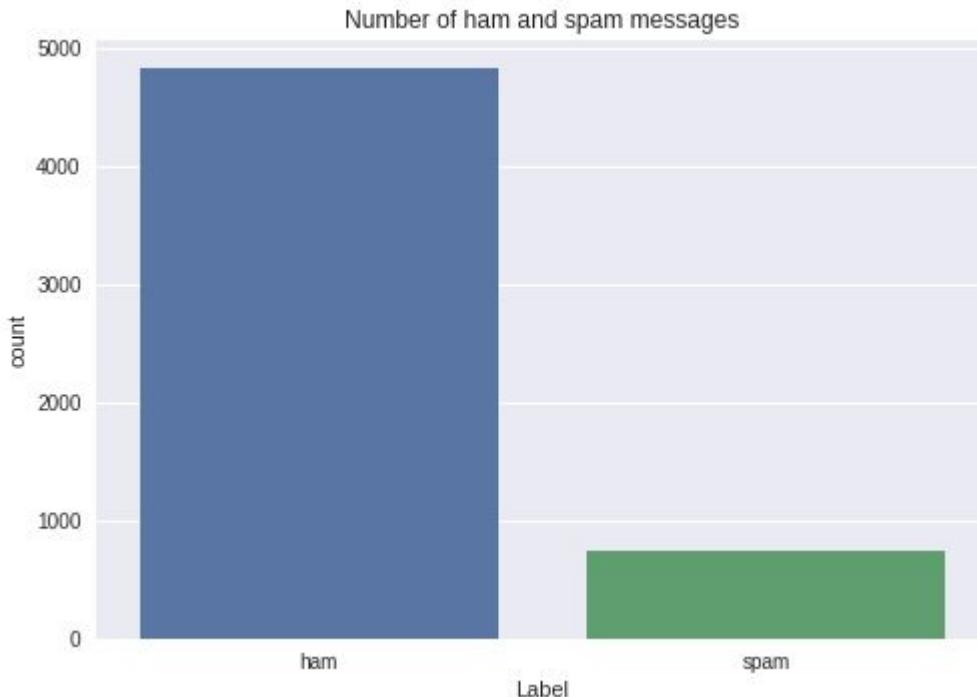
Notebook Address <https://colab.research.google.com/drive/1mC8xs1J7nGlpDbCt2oCuFcsfGJ545QiM>



Workflow

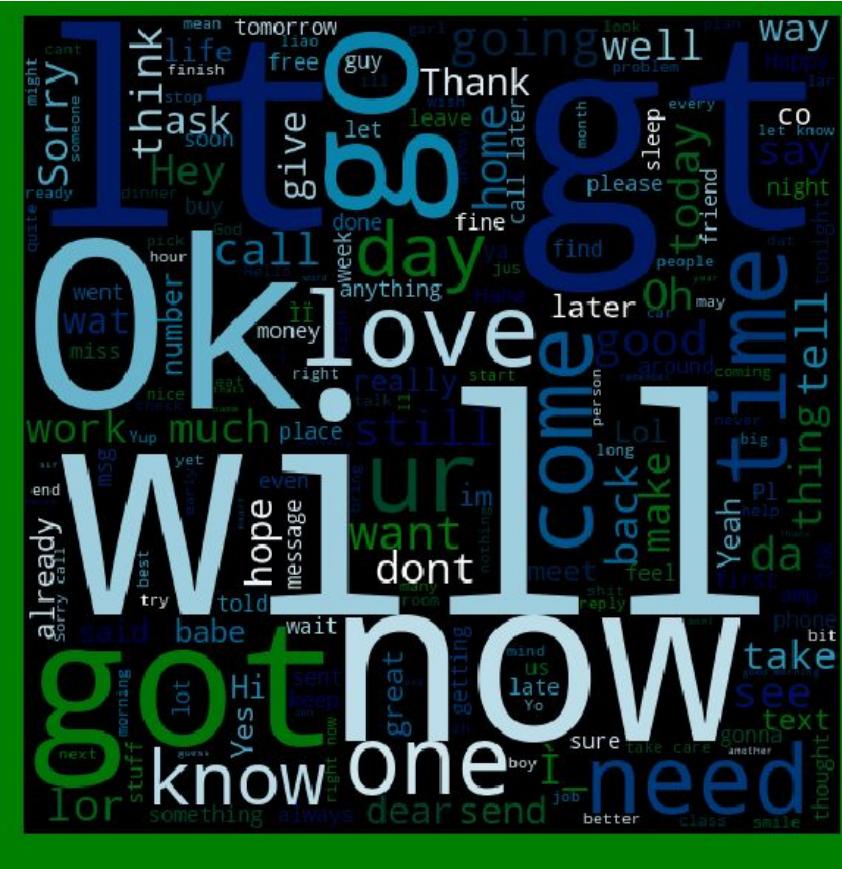


SPAM vs HAM - sms textual data



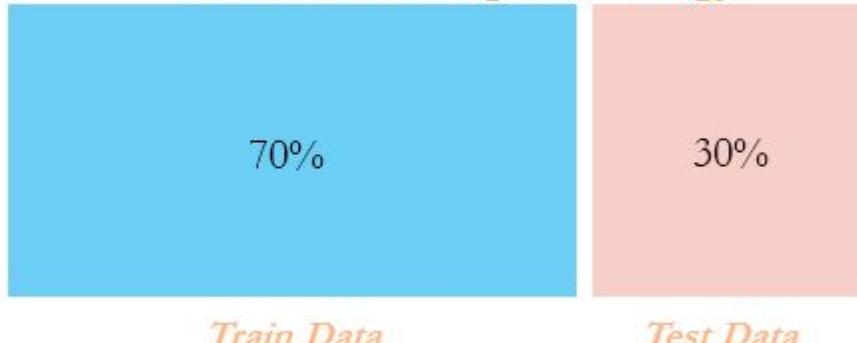
Label	Text
0 ham	Go until jurong point, crazy.. Available only ...
1 ham	Ok lar... Joking wif u oni...
2 spam	Free entry in 2 a wkly comp to win FA Cup fina...
3 ham	U dun say so early hor... U c already then say...
4 ham	Nah I don't think he goes to usf, he lives aro...
5 spam	FreeMsg Hey there darling 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...

SPAM vs HAM - exploratory analysis

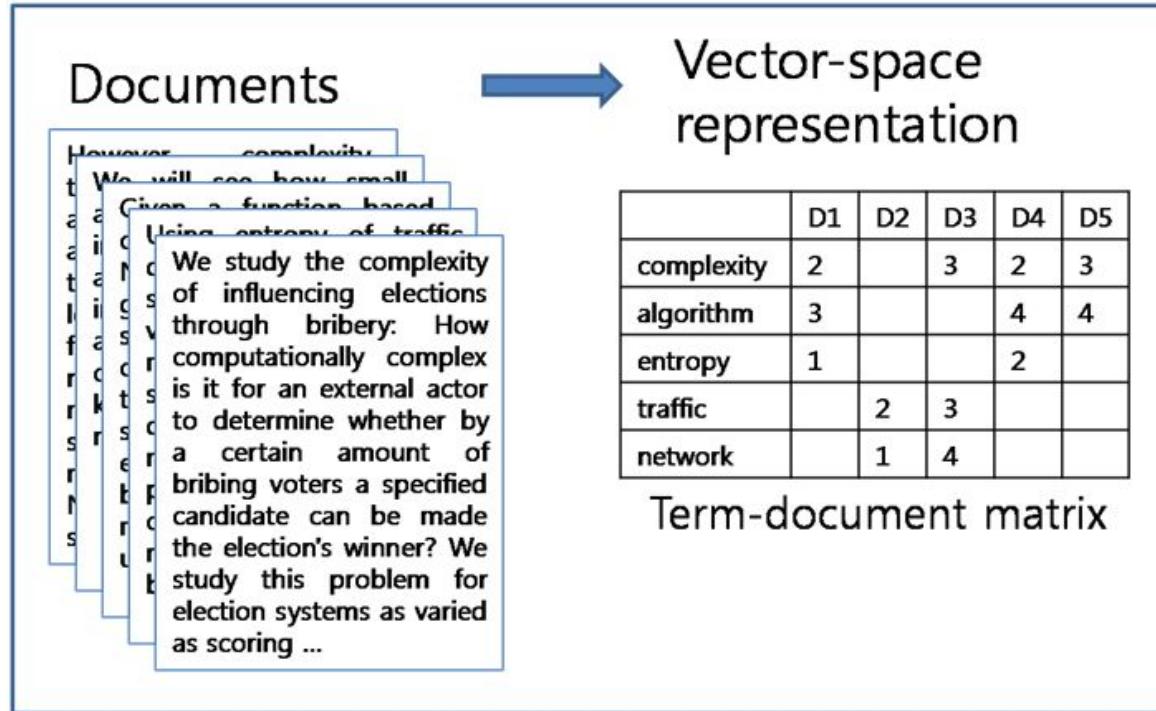


Train / Test Split

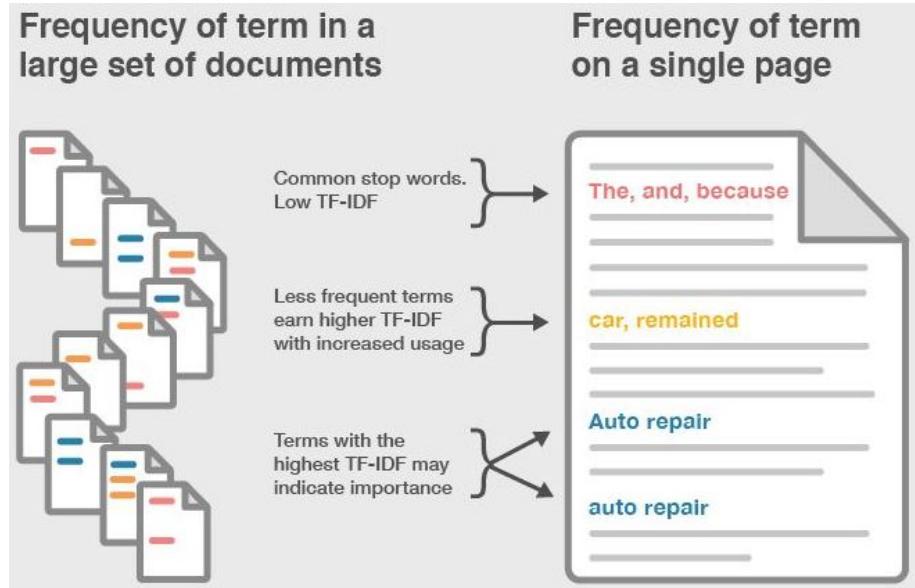
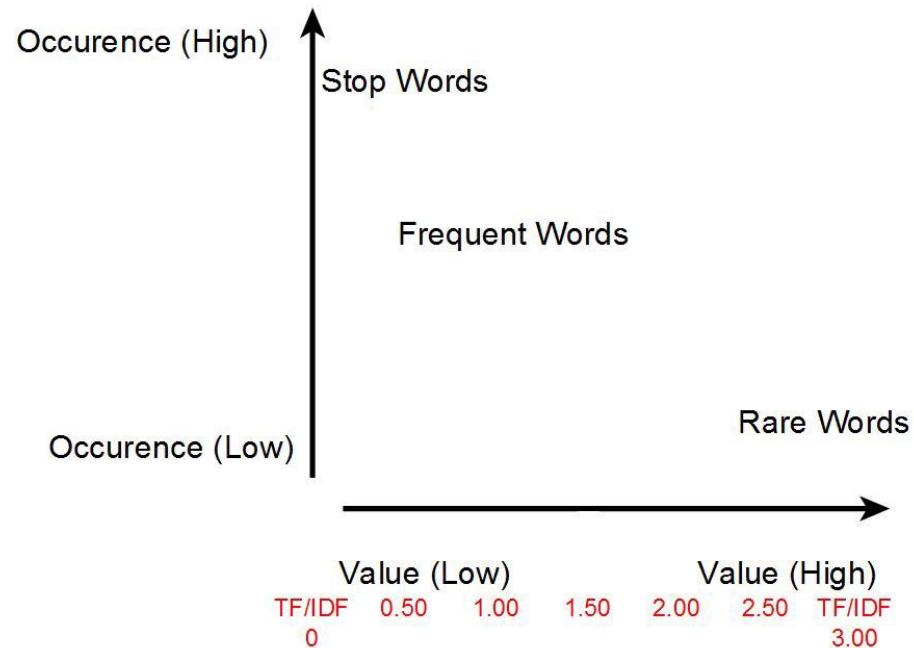
Statistical Modeling Methodology



CountVectorizer creates DTM Document Term Matrix



TF-IDF Term Frequency Inverse Document Frequency



NB - Naive Bayes (1)



“Buy”

25 Spam



75 No spam



$P(\text{“Buy” is Spam}) = ?$

NB - Naive Bayes (2)



“Cheap”

Spam



No spam



$$P(\text{“Cheap” is Spam}) = ?$$

NB - Naive Bayes (3)



“Buy” and “Cheap”

Spam

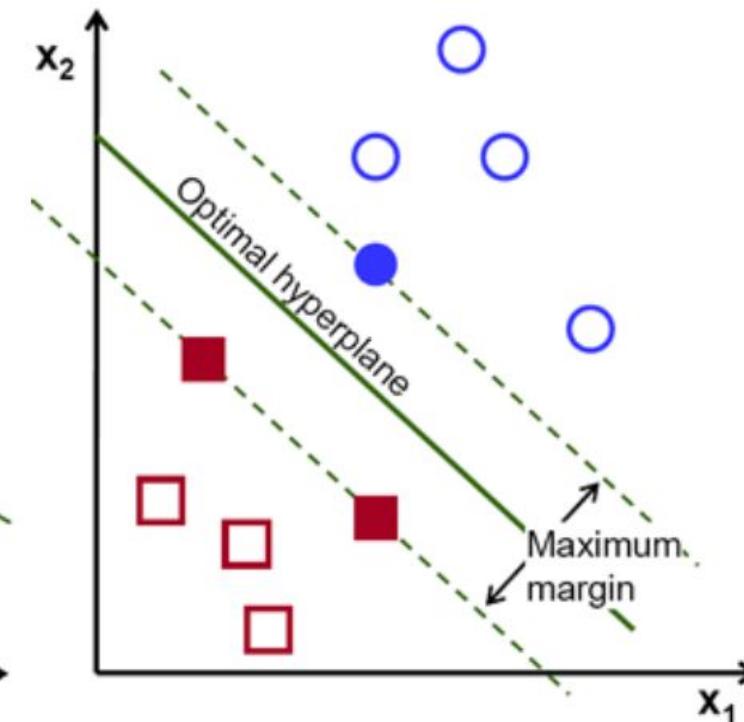
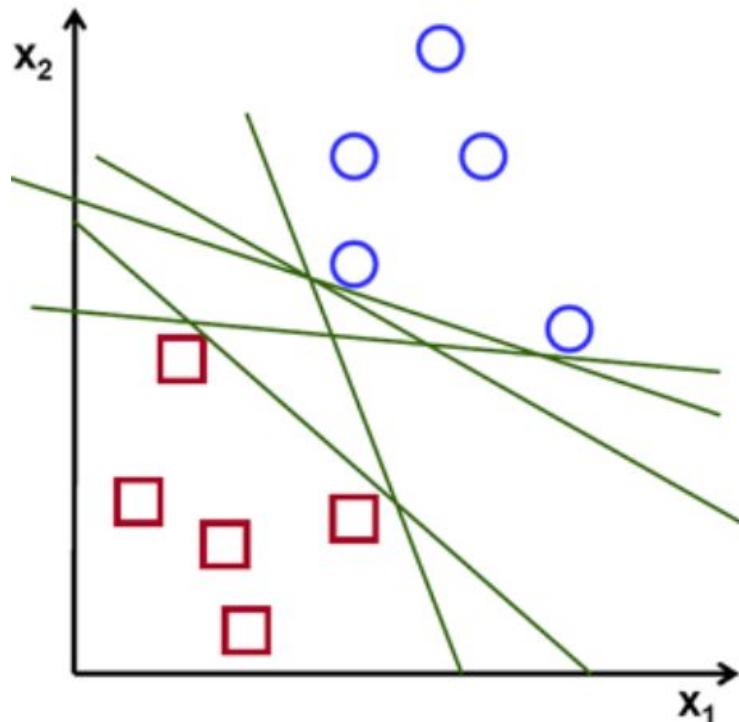


No spam

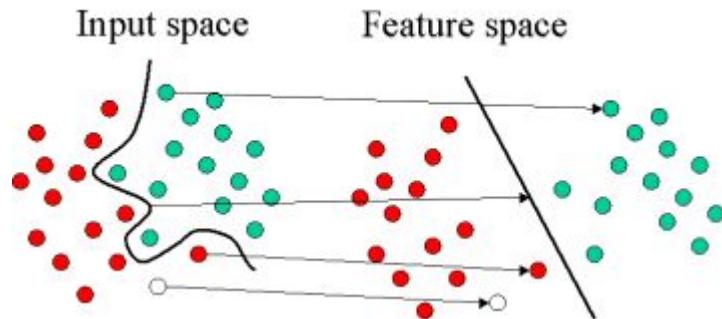


$P(\text{“Buy”} \& \text{“Cheap”} \text{ is Spam}) = ?$

SVM - Support Vector Machine

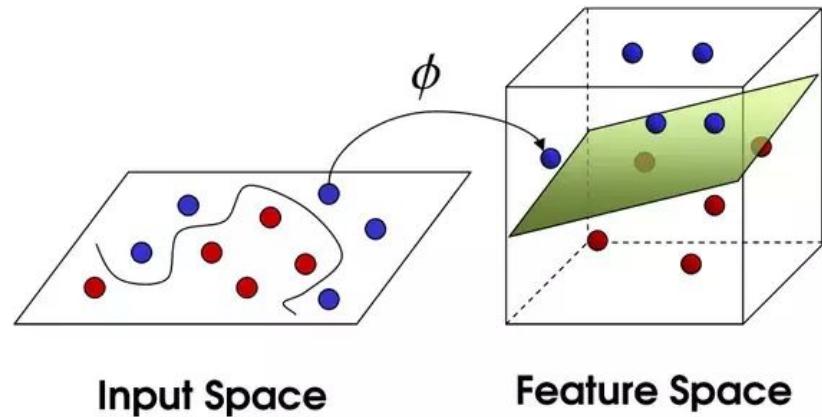


SVM - Tuning Hyper Parameters

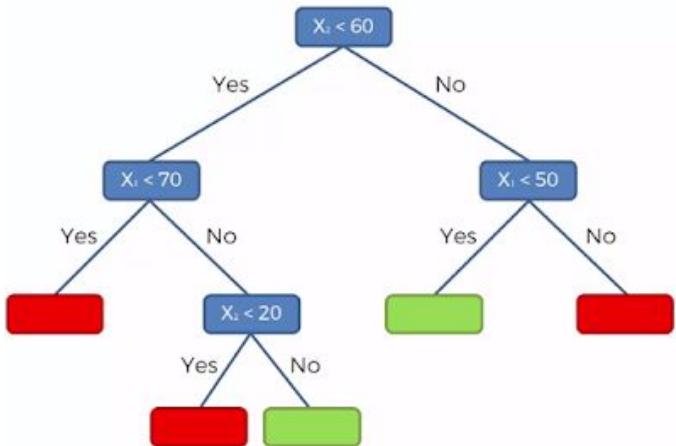
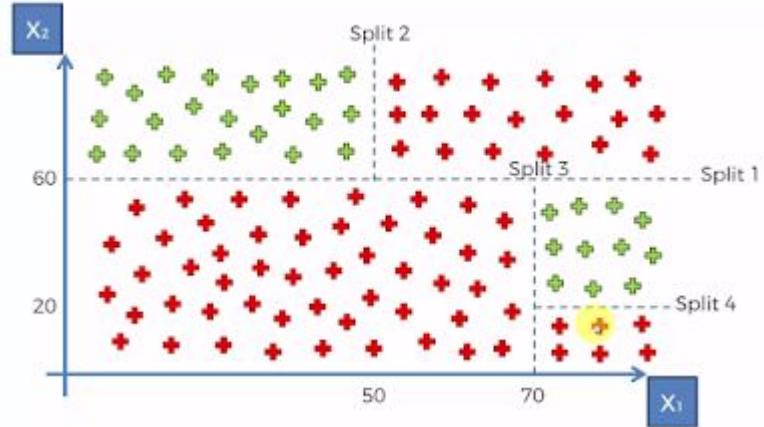
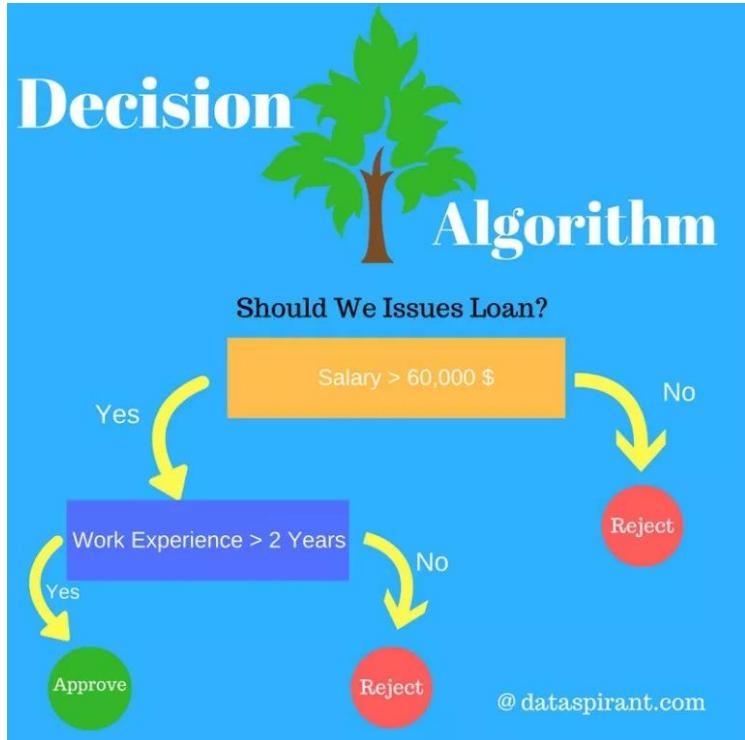


ϕ - is mapping
between
Non-Linear and
Linear Separable

SVM Kernel



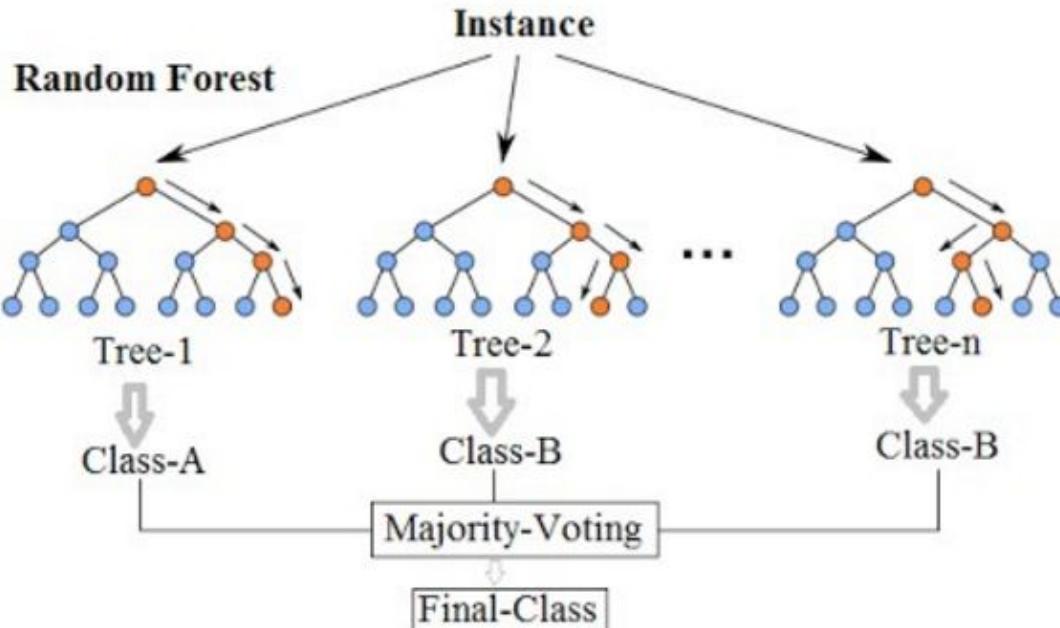
DT - Decision Tree



GROUPON®

Random Forest - many DT and Voting

Random Forest Simplified



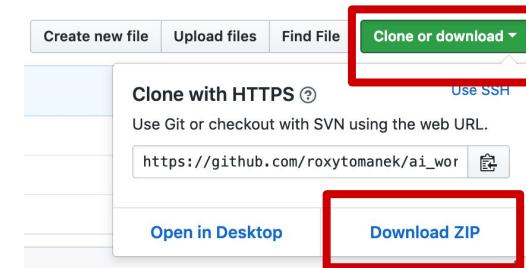
WORKSHOP

Automatic Document Categorization with Machine Learning techniques in Python

[https://github.com/
roxytomanek/ai_workshop](https://github.com/roxytomanek/ai_workshop)

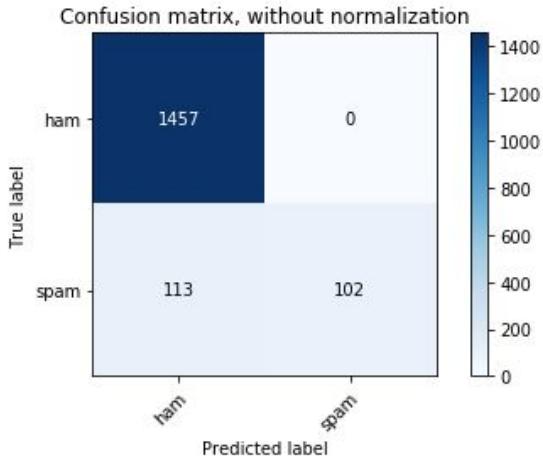
(All data and information)

- 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'

Task to do... Accuracy & Computation Time



Naive
Bayes

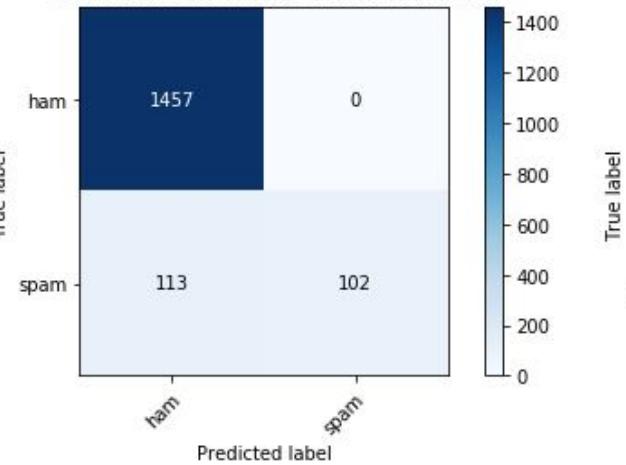


Random
Forest



Support
Vector
Machine

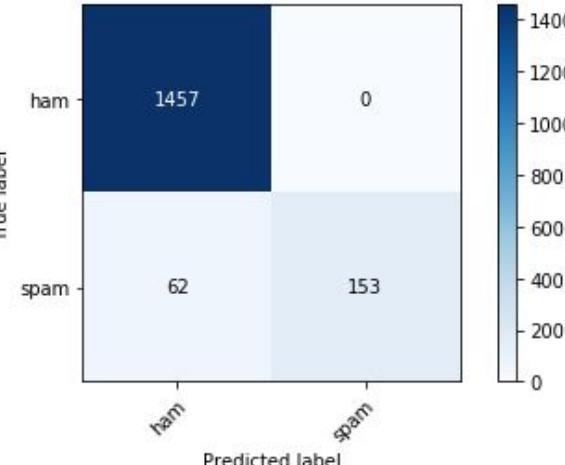
Confusion matrix, without normalization



NB

Training done in 0.366s
Model final score: 0.932

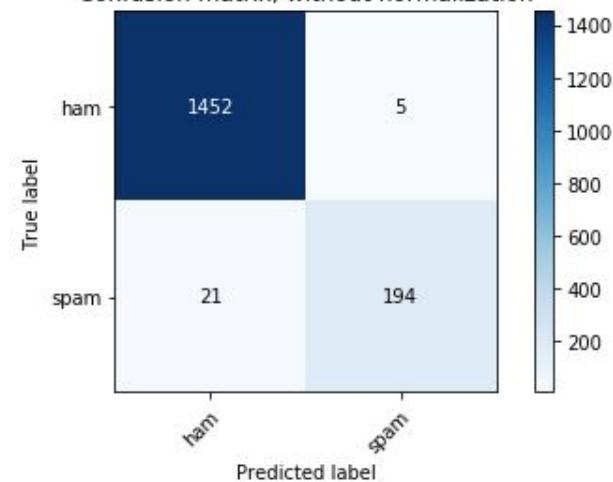
Confusion matrix, without normalization



RF

Training done in 0.480s
Model final score: 0.963

Confusion matrix, without normalization

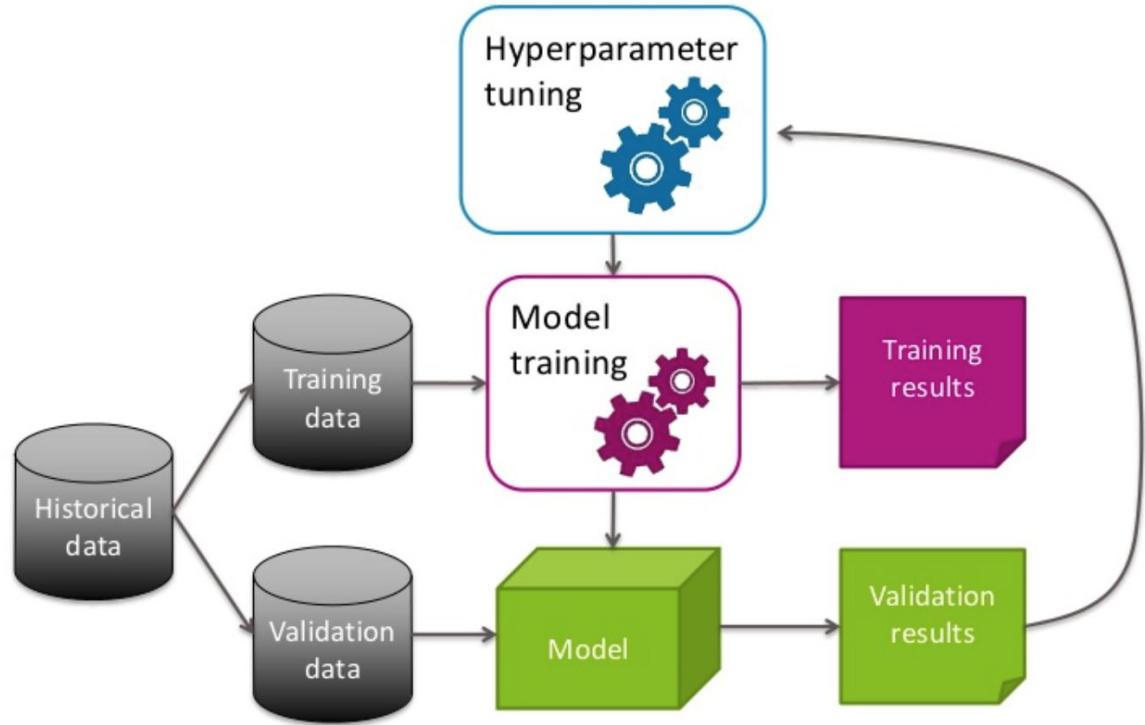


SVM

Training done in 5.247s
Model final score: 0.984

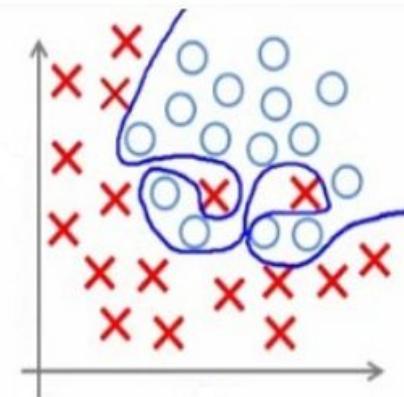
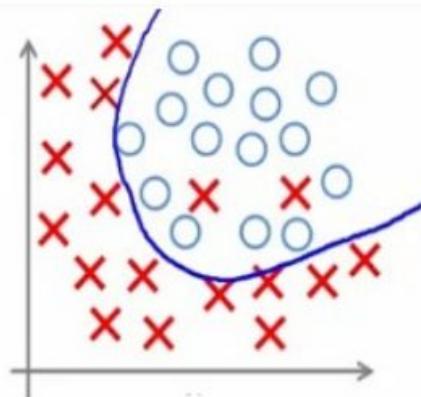
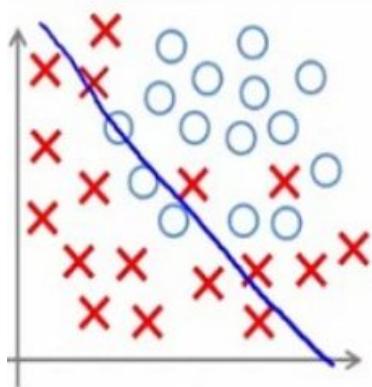
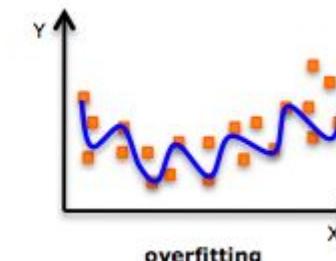
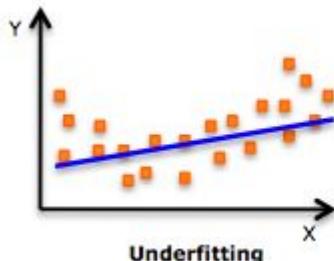
GROUPON®

Model Selection and Tuning

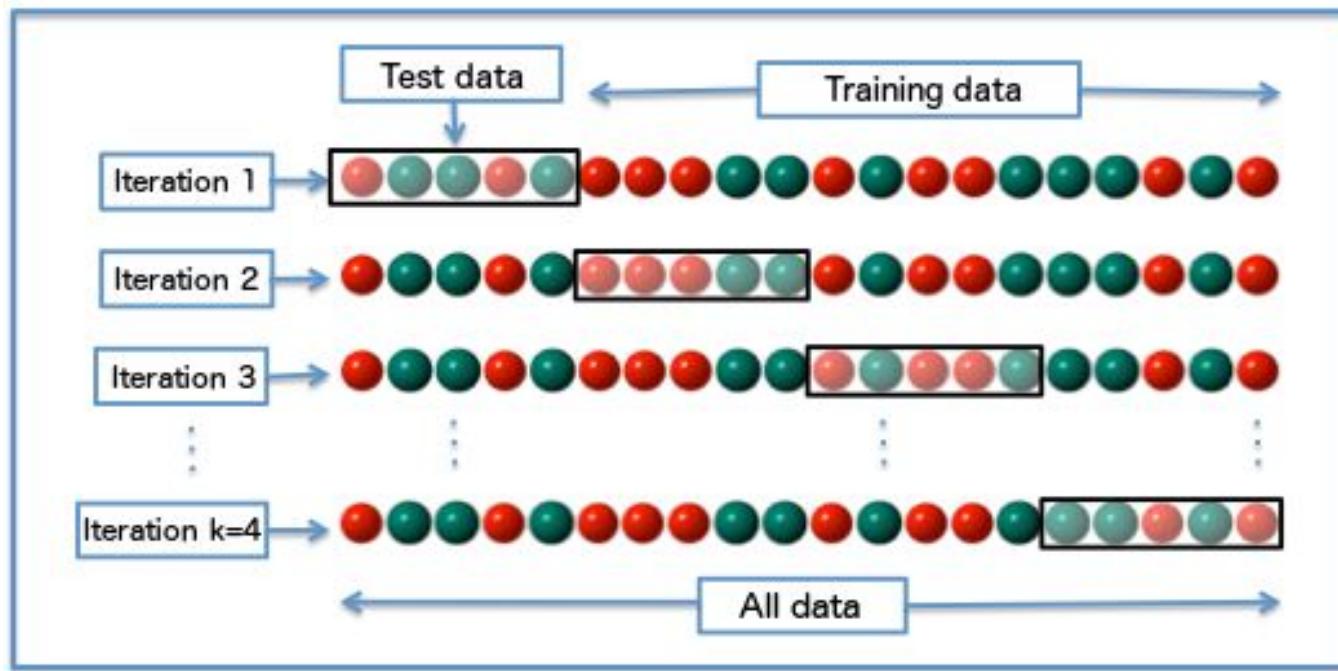


<https://www.slideshare.net/AliceZheng3/evaluating-machine-learning-models-a-beginners-guide>

Underfit and Overfit



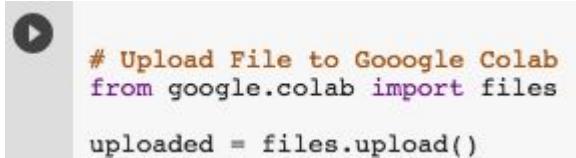
Against Overfit - Cross-Validation



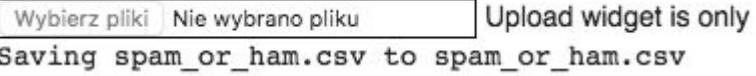
4-Fold Cross-Validation

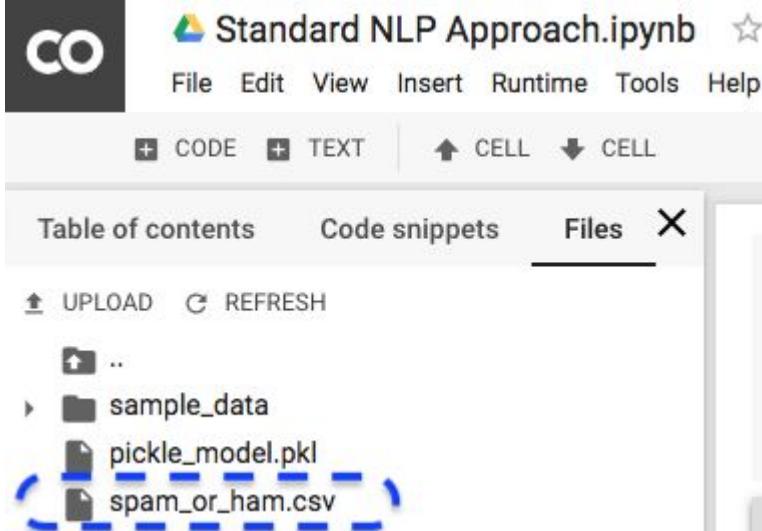
GROUPON®

upload file for 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 

CO Standard NLP Approach.ipynb ⭐

File Edit View Insert Runtime Tools Help

+ CODE + TEXT ↑ CELL ↓ CELL

Table of contents Code snippets Files X

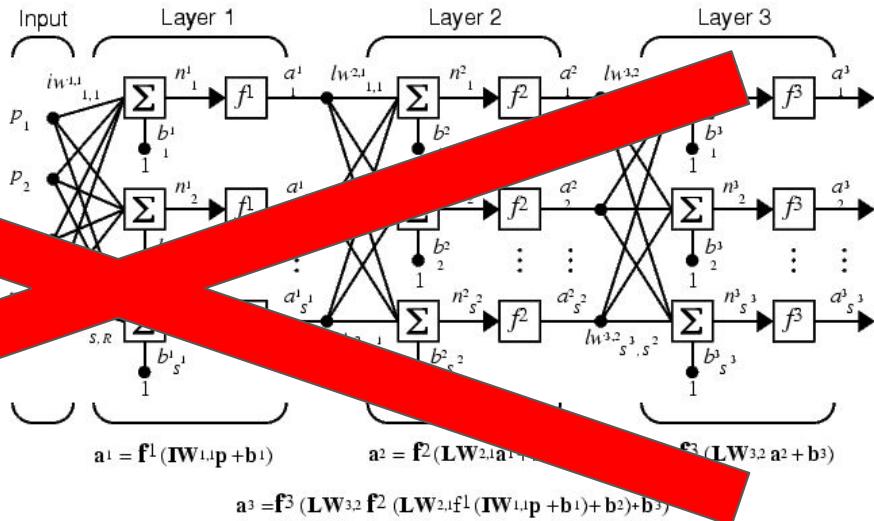
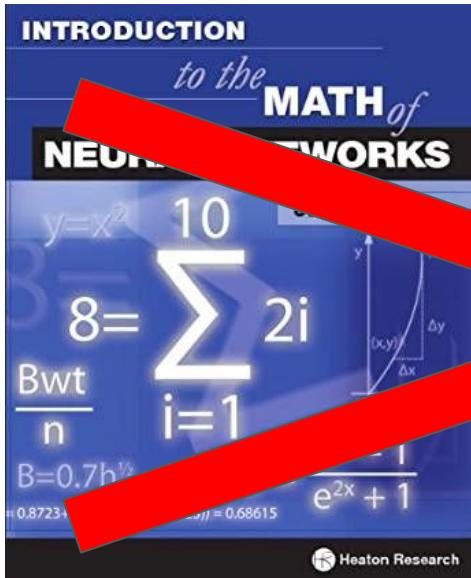
UPLOAD REFRESH

 - ..
 - sample_data
 - pickle_model.pkl
 - spam_or_ham.csv

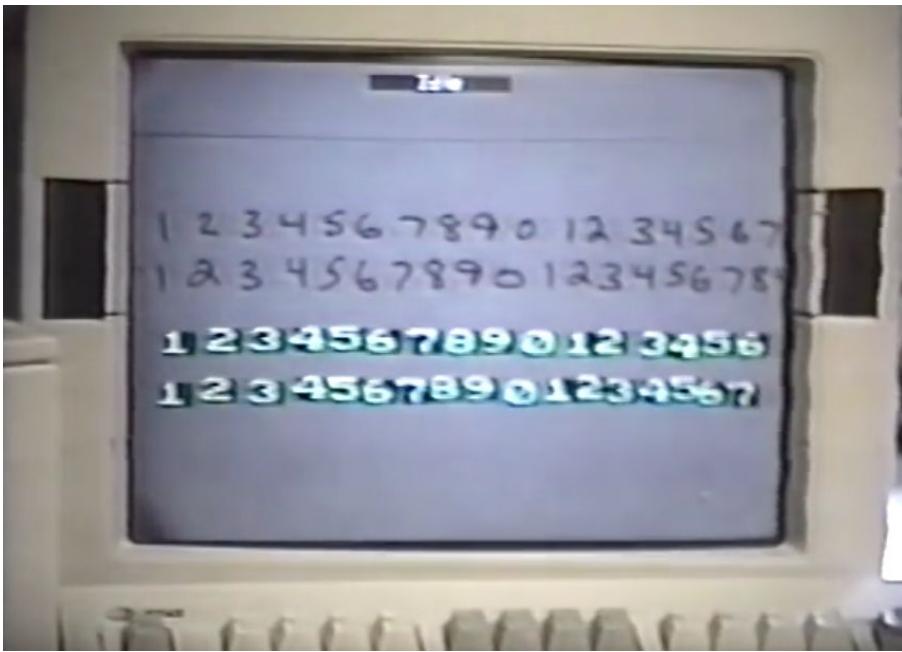
Part 2

Neural Networks

Neural Networks University Approach



MNIST - Digit Recognition



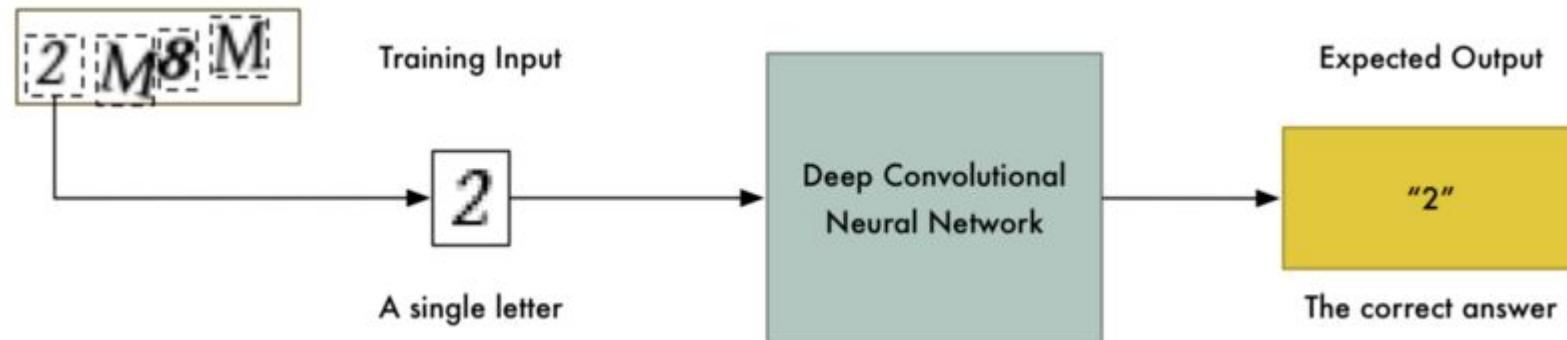
000000000000000000
111111111111111111
222222222222222222
333333333333333333
444444444444444444
555555555555555555
666666666666666666
777777777777777777
88888888888888888888
99999999999999999999

Convolutional Network Demo from 1993

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

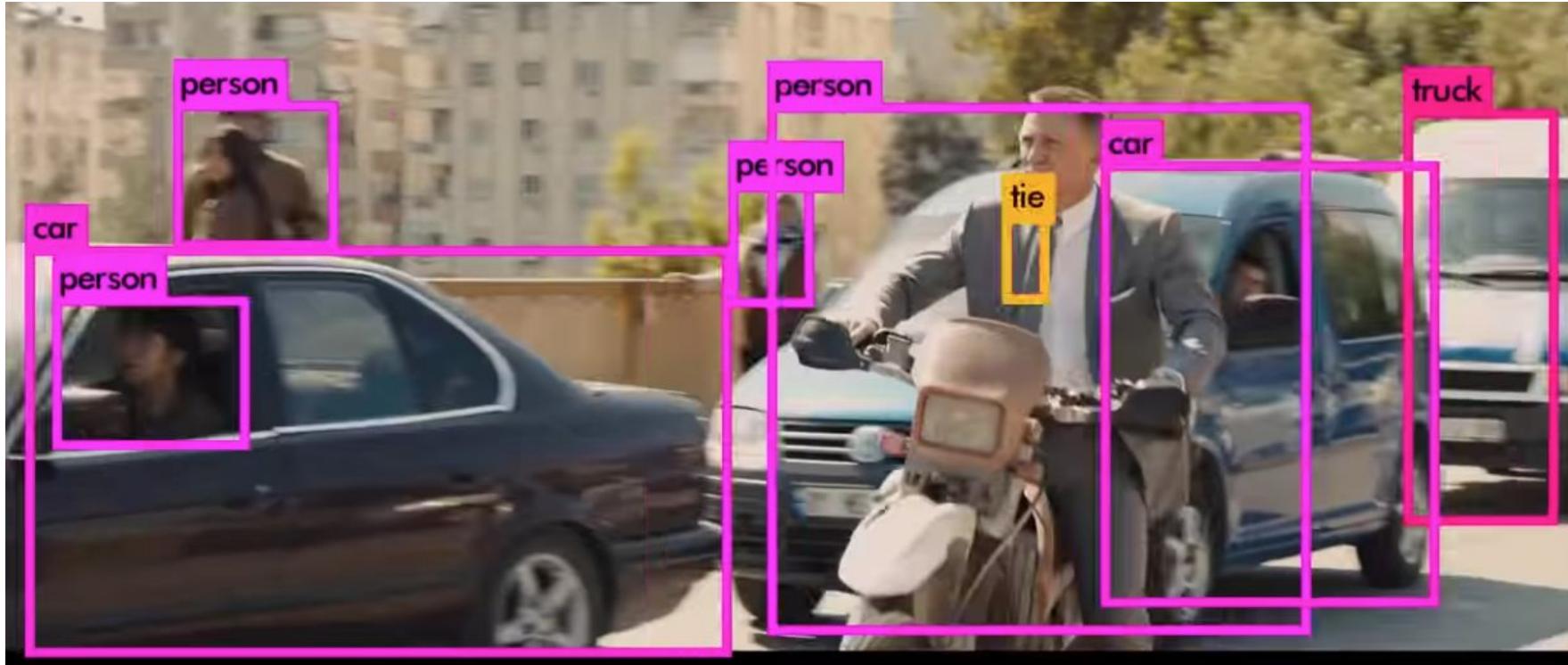
Groupon®

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>

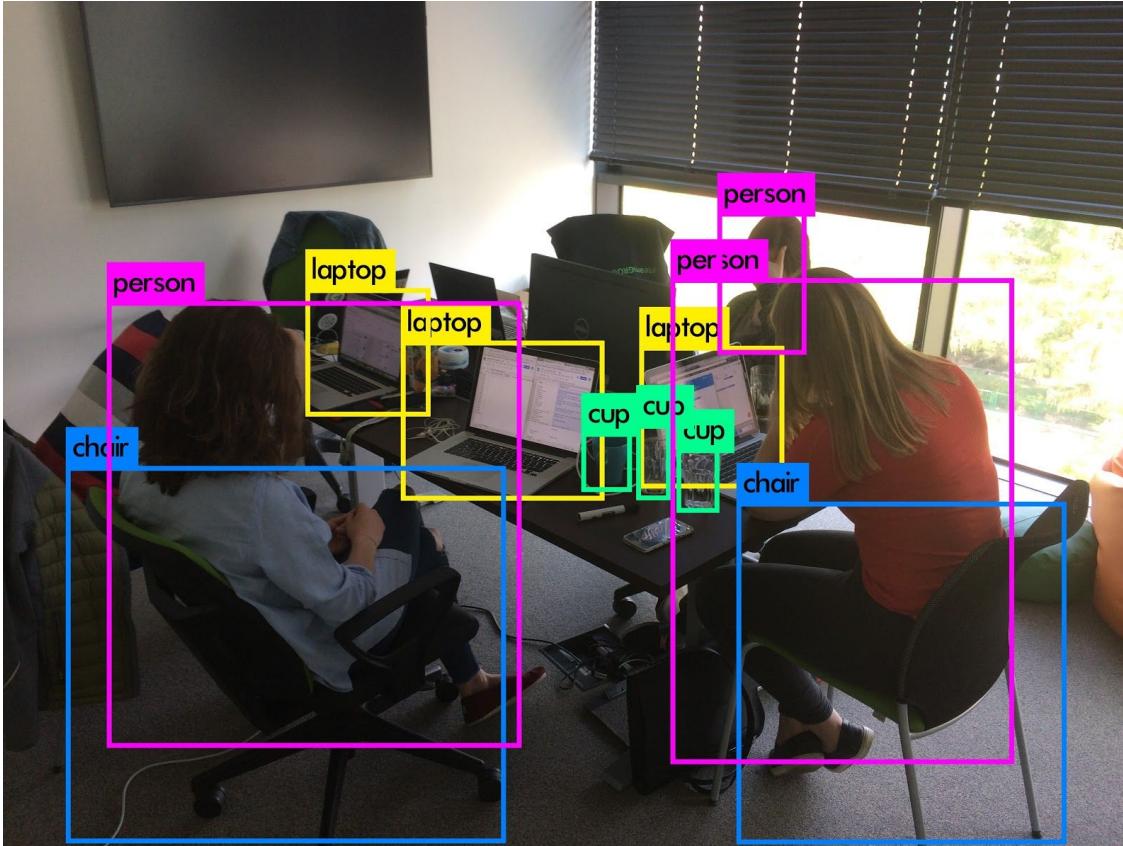
Why Neural Networks ?



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

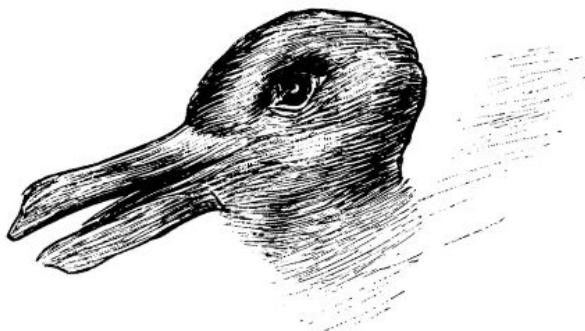
TED | Joseph Redmon <https://www.youtube.com/watch?v=CgxsvTriJhl> Groupon®

YOLO in Groupon Katowice

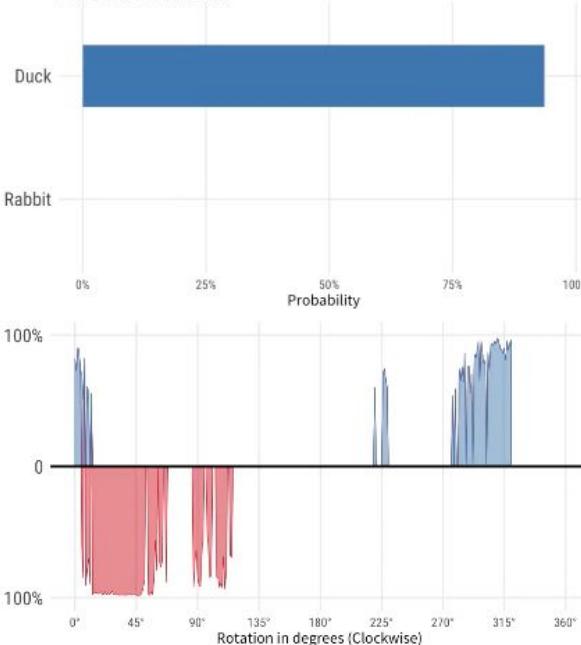


GROUPON®

Hacking CNN, Duck or Rabbit



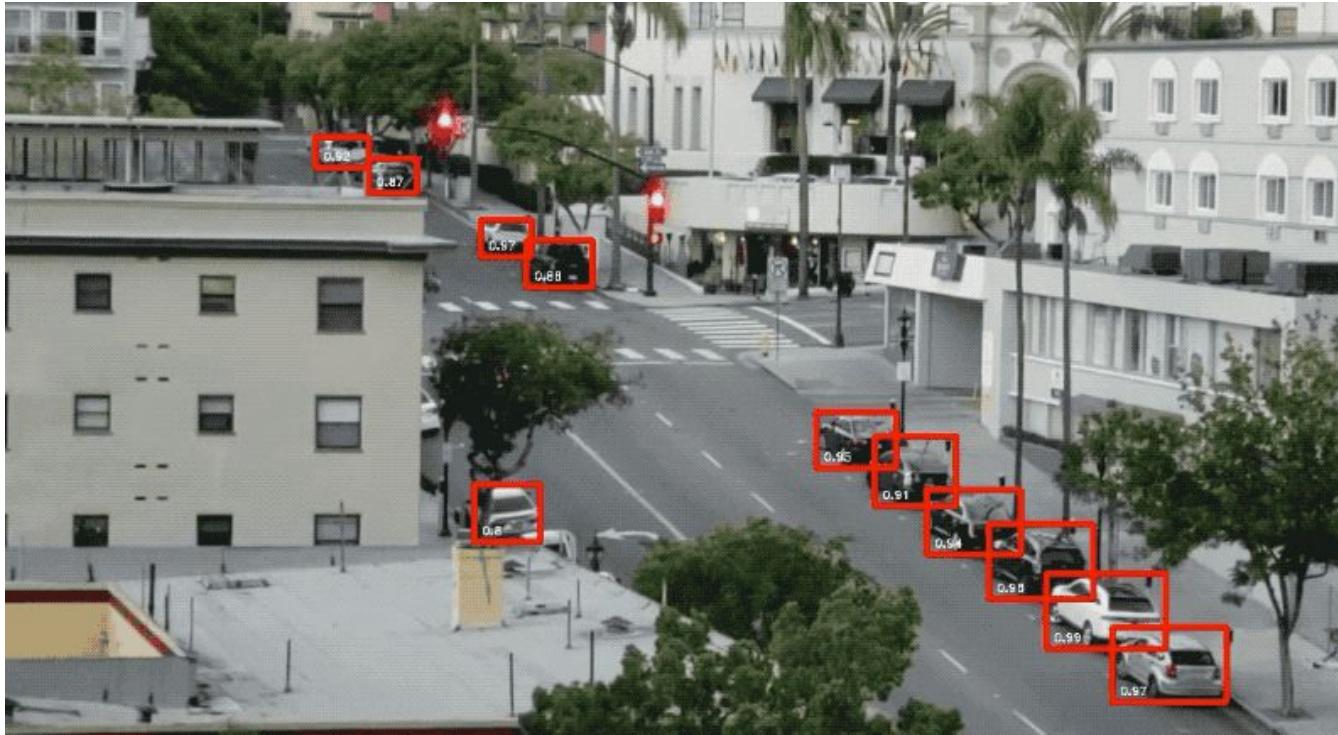
Duck or Rabbit?



Predictions provided by the Google Cloud Vision API
Animation by Max Woolf (@minimaxir)

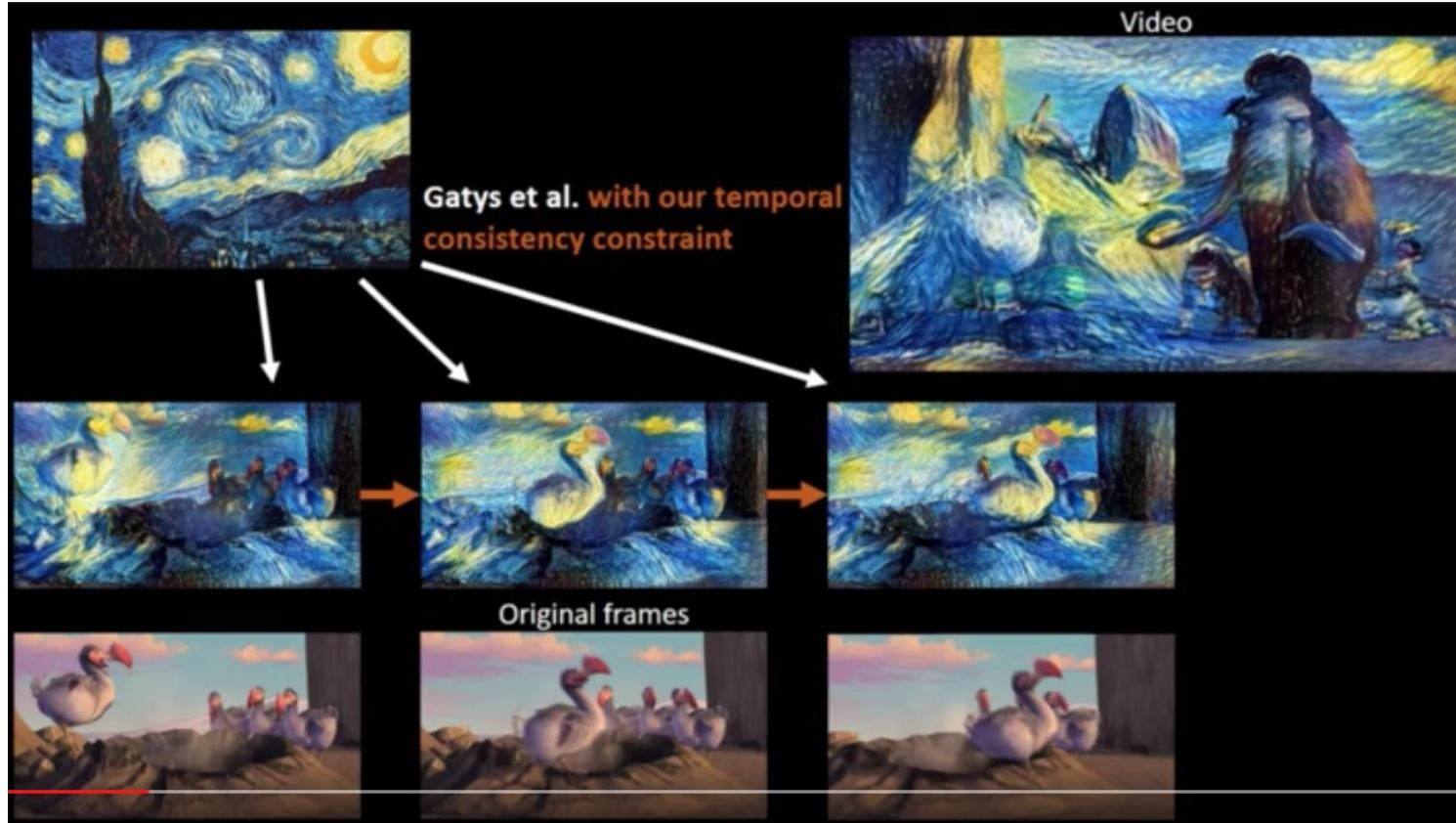
https://www.reddit.com/r/dataisbeautiful/comments/aydqig/is_it_a_duck_or_a_rabbit_for_google_cloud_vision

Snagging Parking Spaces with Mask R-CNN and Python



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

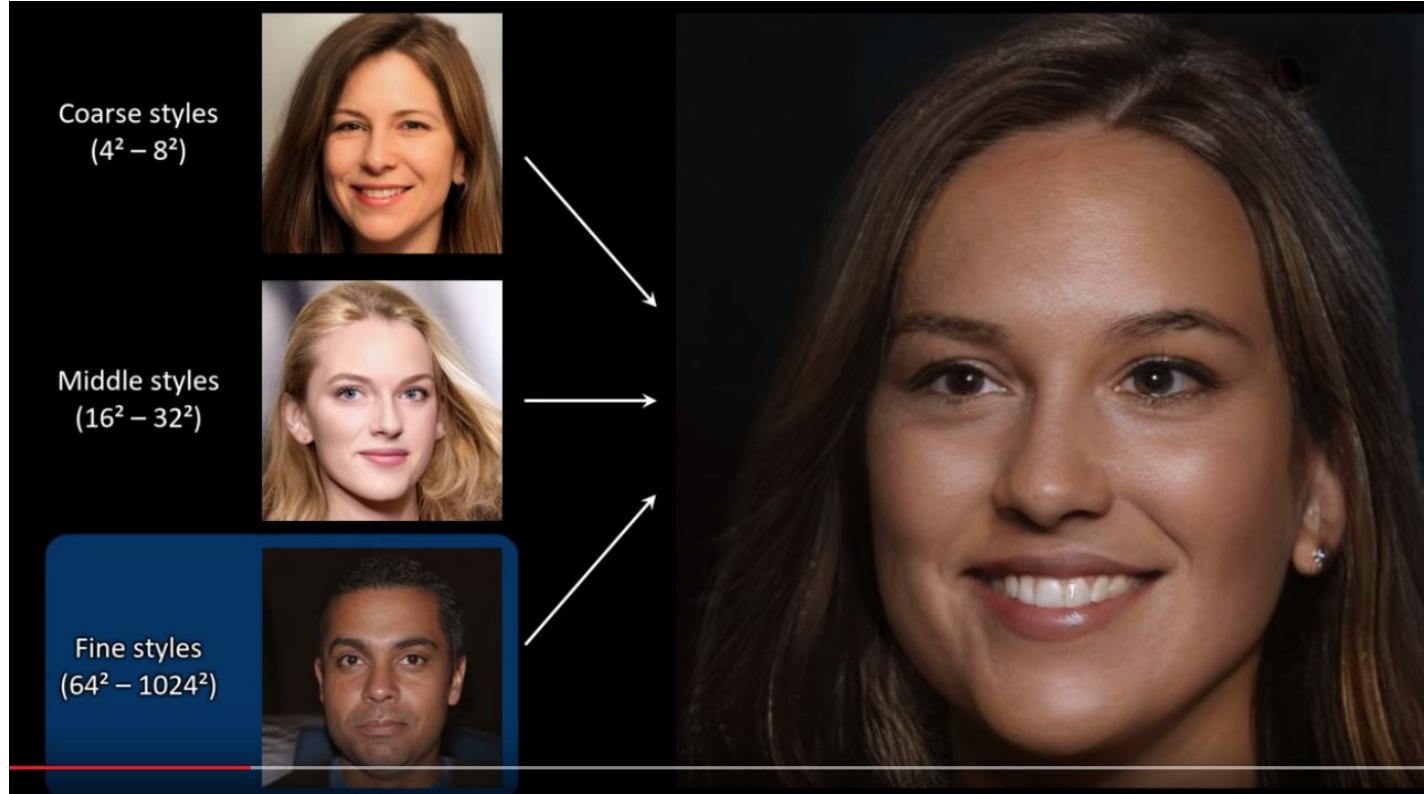
Artistic style transfer for videos



<https://www.youtube.com/watch?v=Khuj4ASIdmU>

GROUPON®

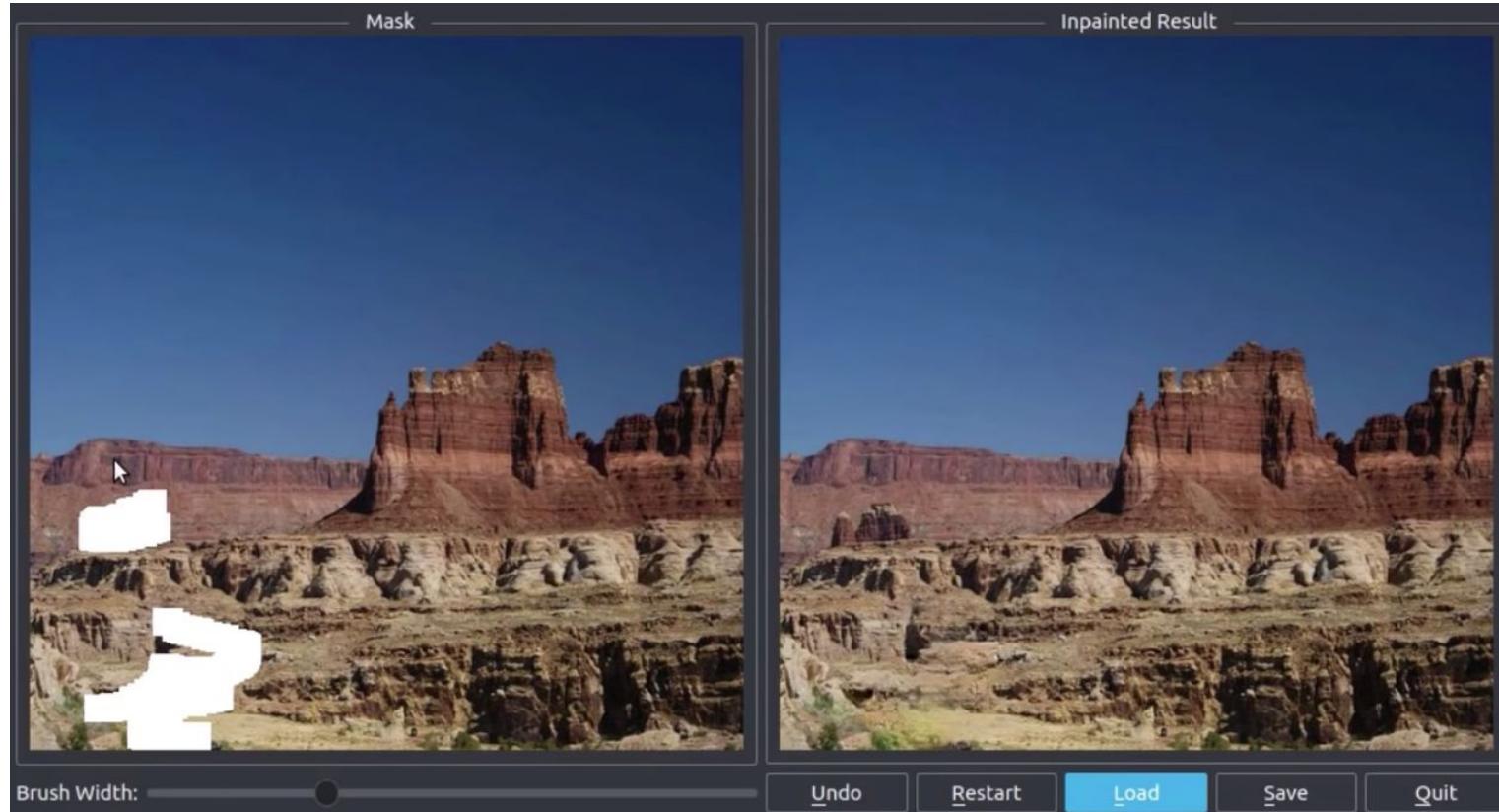
A Style-Based Generator Architecture for Generative Adversarial Networks



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

GROUPON®

AI Reconstructs Photos with Realistic Results



<https://www.youtube.com/watch?v=gg0F5jKmhA>

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

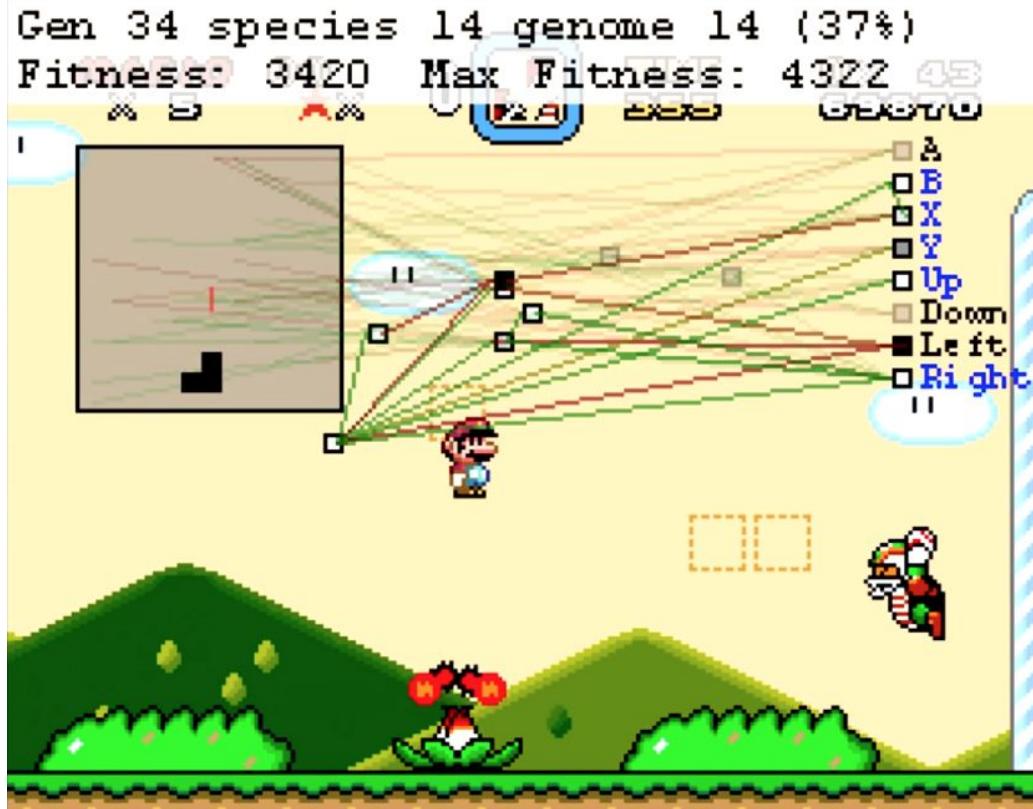
<https://www.forbes.com/sites/williamfalcon/2018/10/25/what-happens-now-that-an-ai-generated-painting-sold-for-432500>

GROUPON®

GAN in Groupon



Neural Networks in Games



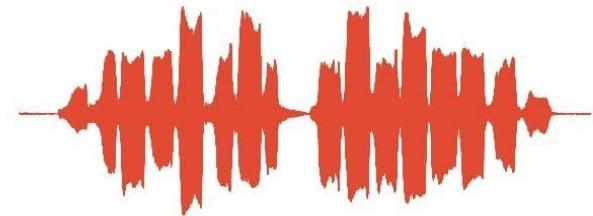
NEAT

Neuroevolution
of
Augmenting
Topologies

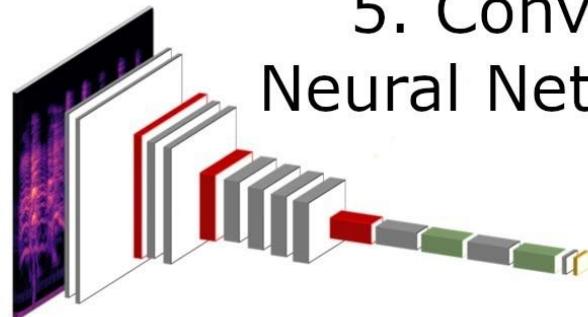
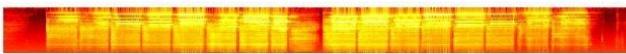
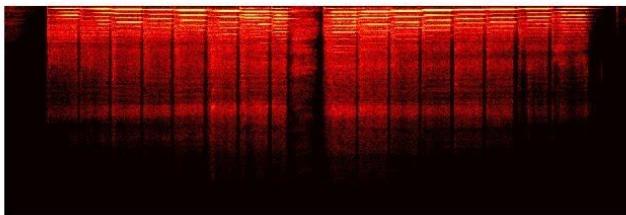
Mari/O - Machine Learning for Video Games
<https://www.youtube.com/watch?v=qv6UVOQ0F44>

GROUPON®

Deep Learning for Audio Classification



Deep Learning
for Audio
Classification



5. Conv
Neural Net

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

GROUPON®

Deep Learning SR + TTS



“Hi, I'm calling to book a women's haircut for a client.”



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

©2016 Fjodor van Veen - asimovinstitute.org

○ Backfed Input Cell

○ Input Cell

△ Noisy Input Cell

● Hidden Cell

○ Probabilistic Hidden Cell

△ Spiking Hidden Cell

● Output Cell

○ Match Input Output Cell

● Recurrent Cell

○ Memory Cell

△ Different Memory Cell

● Kernel

○ Convolution or Pool

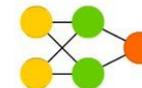
Perceptron (P)



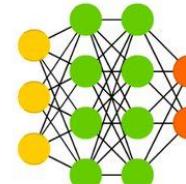
Feed Forward (FF)



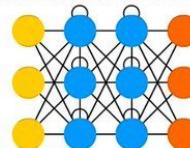
Radial Basis Network (RBF)



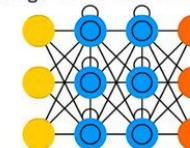
Deep Feed Forward (DFF)



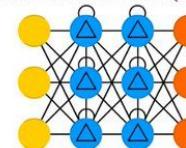
Recurrent Neural Network (RNN)



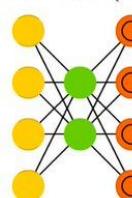
Long / Short Term Memory (LSTM)



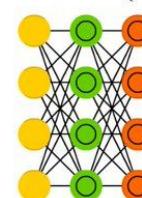
Gated Recurrent Unit (GRU)



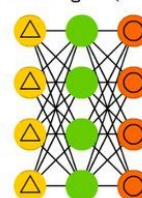
Auto Encoder (AE)



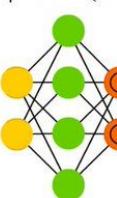
Variational AE (VAE)



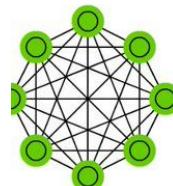
Denoising AE (DAE)



Sparse AE (SAE)



Markov Chain (MC)



Hopfield Network (HN)



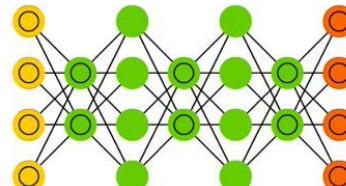
Boltzmann Machine (BM)



Restricted BM (RBM)

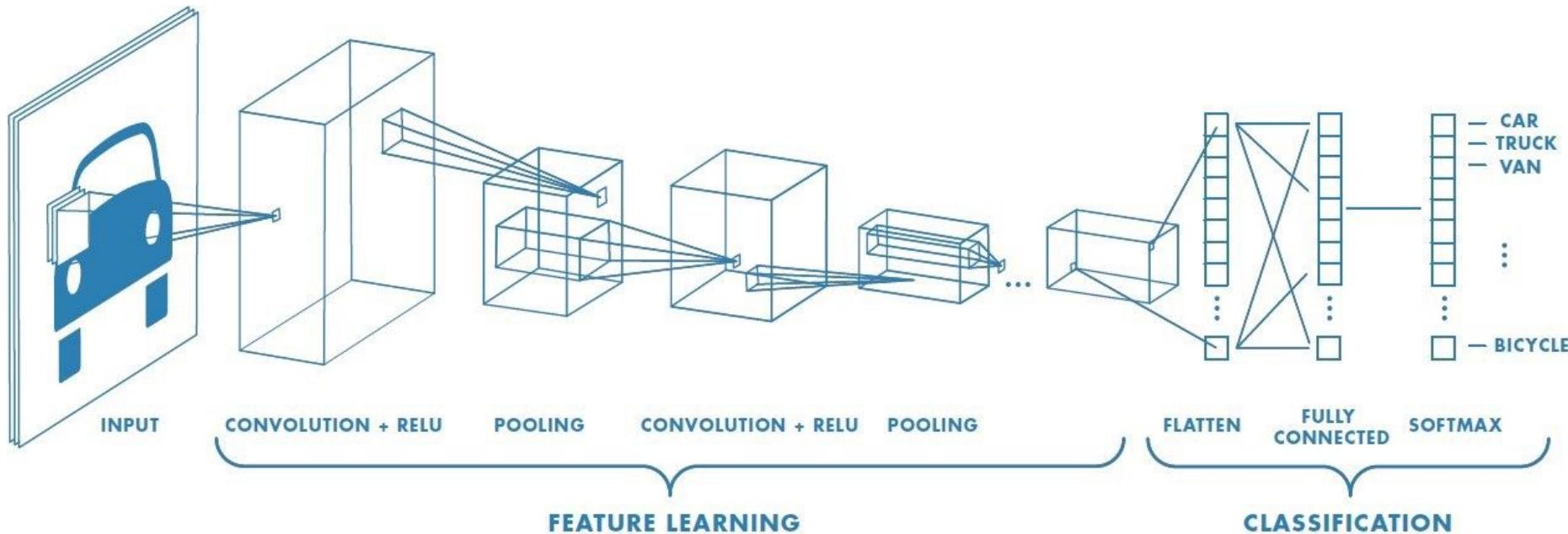


Deep Belief Network (DBN)



GROUPON®

Topology Convolutional Network



Keras Model



TensorFlow

```
# 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

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', '.']
```

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 sentences 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
  1 60  8 16  3 83]
```

Keras Model

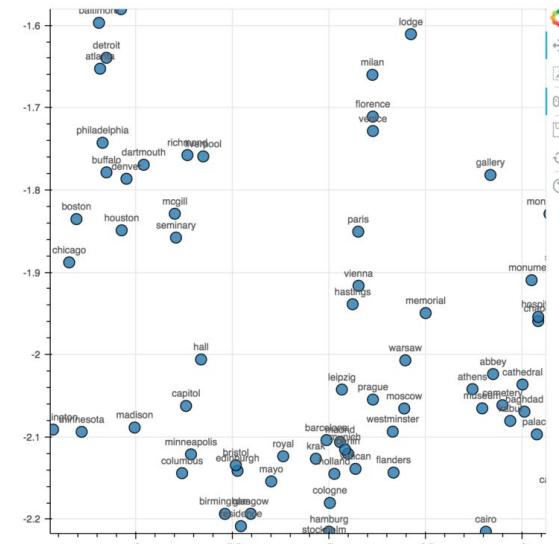
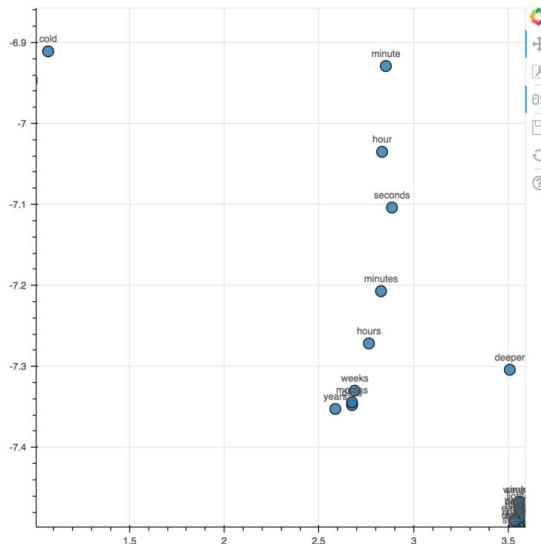
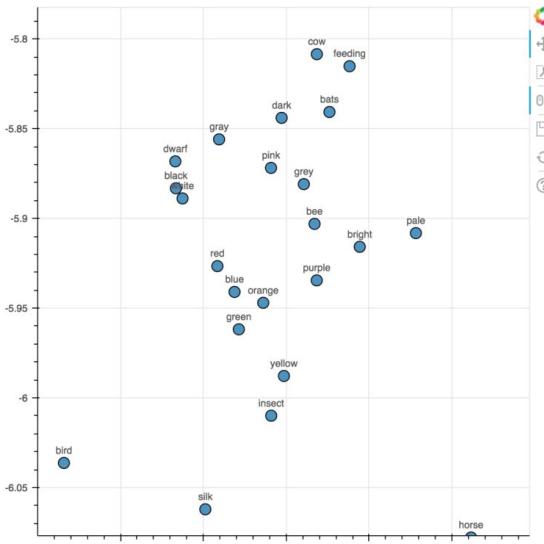
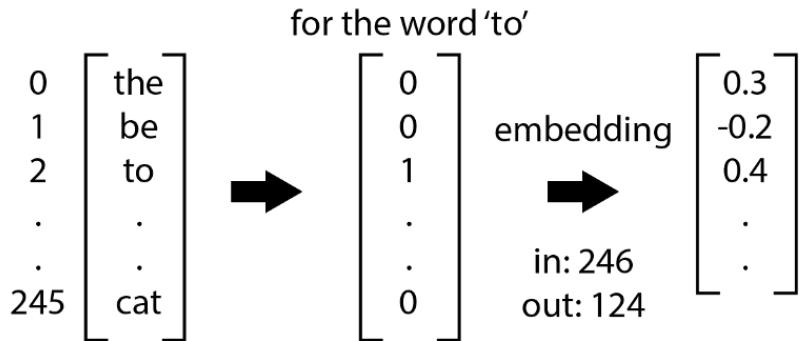


TensorFlow

```
# 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

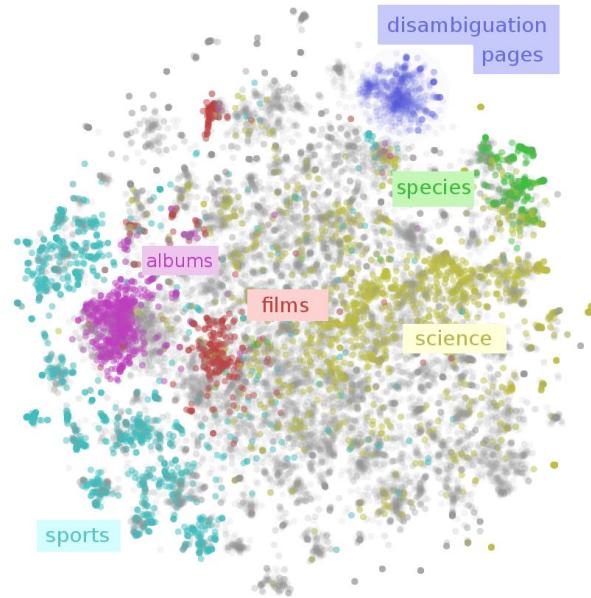


<https://nlpforhackers.io/word-embeddings>

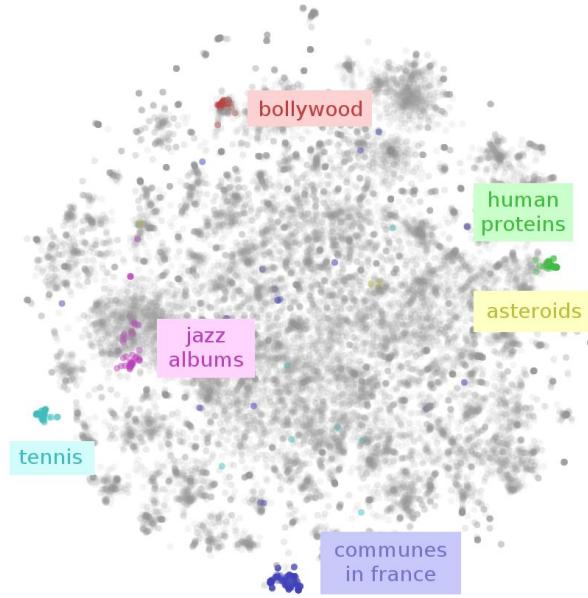
GROUPON®

Map of Wikipedia (t-SNE)

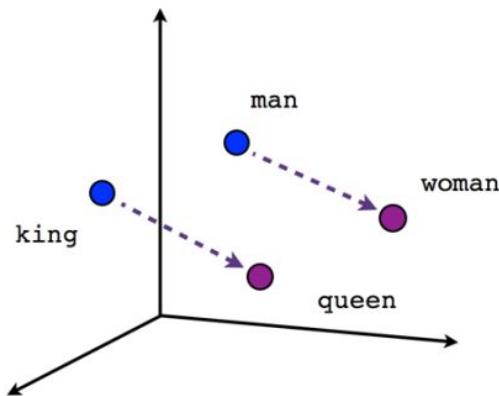
Large Clusters



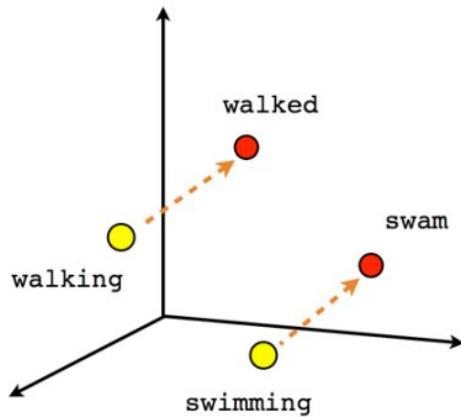
Small Clusters



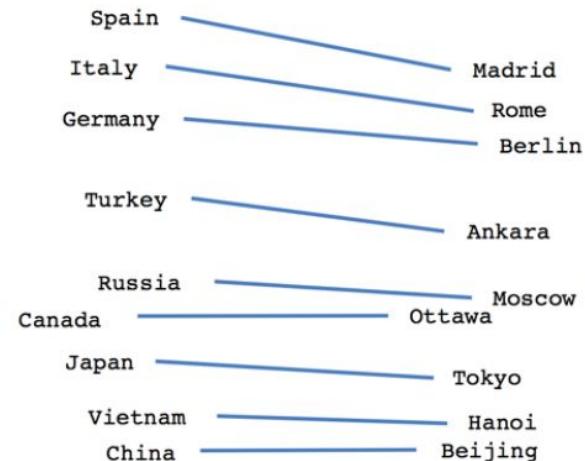
Word2Vec - Hidden Semantic Relationships



Male-Female



Verb tense



Country-Capital

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

GROUPON®

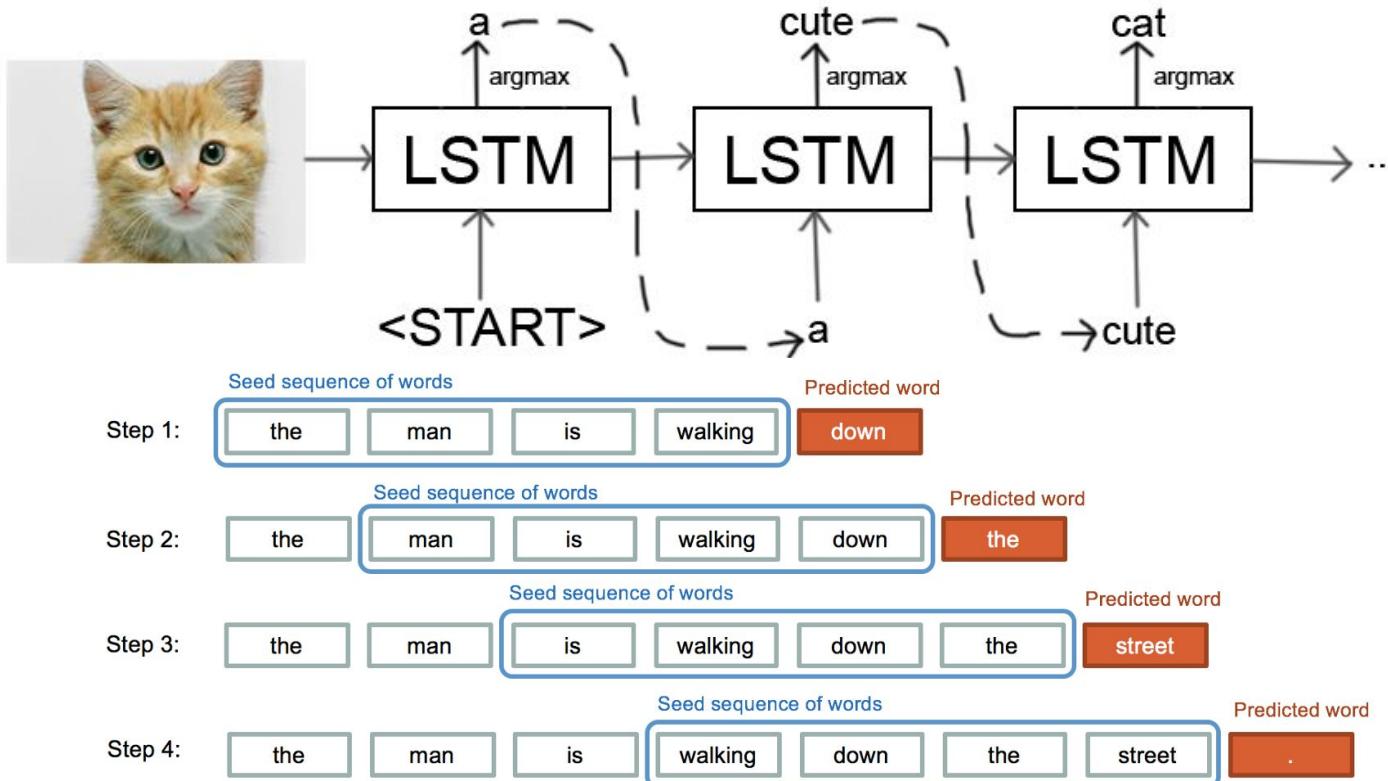
Keras Model



TensorFlow

```
# 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>

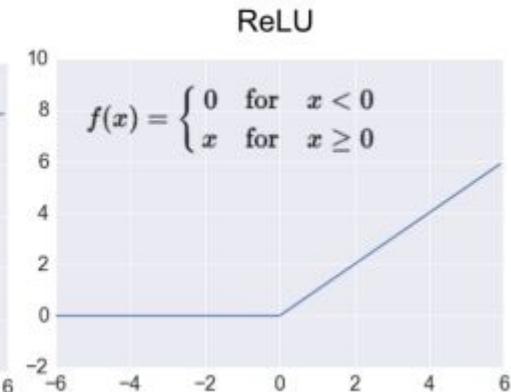
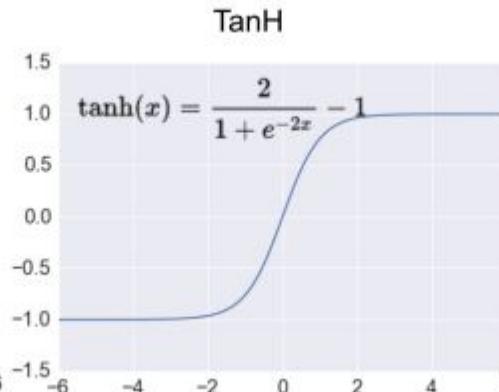
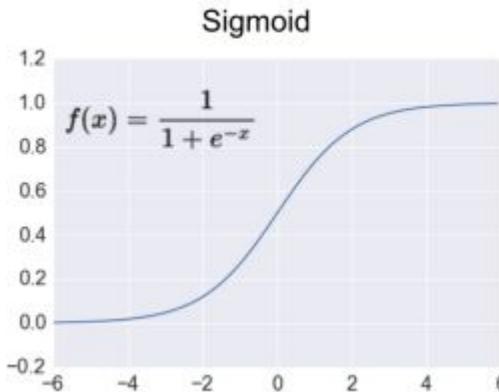
Keras Model



TensorFlow

```
# 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>

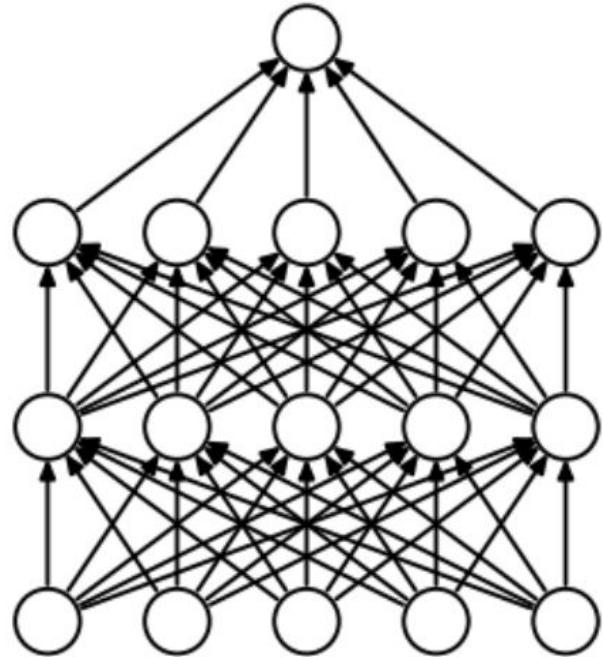
Keras Model



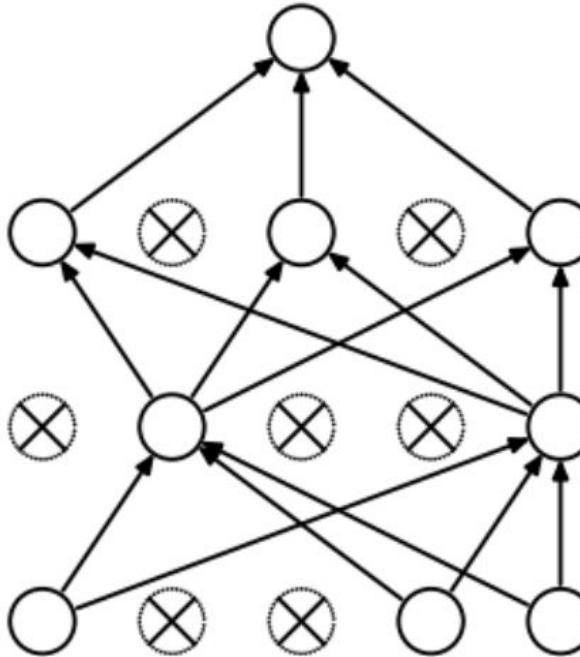
TensorFlow

```
# 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.

Keras Model



TensorFlow

```
# 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

$$F(X_i) = \frac{\text{Exp}(X_i) \quad i = 0, 1, 2, \dots k}{\sum_{j=0}^k \text{Exp}(X_j)}$$

dataaspirant.com



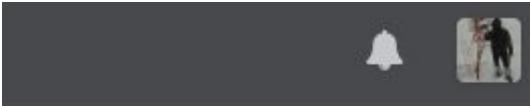
Sigmoid Function

$$F(X_i) = \frac{1}{1 + \text{Exp}(-X_i)}$$

dataaspirant.com

	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.

Kaggle (codes)



Register with just one click:
We won't share anything without your permission

Sign up with Google

Sign up with Facebook

Manually create an account:

Email

Password

Register

The screenshot shows a Kaggle notebook interface. The title is "Standard NLP versus Neural Networks". The code in the editor is:

```
[ ]: # Import the necessary libraries
import numpy as np
import pandas as pd # Dataframe Management
import matplotlib.pyplot as plt
import seaborn as sns # Visualization
from sklearn.model_selection import train_test_split
import pickle # Model Serialization

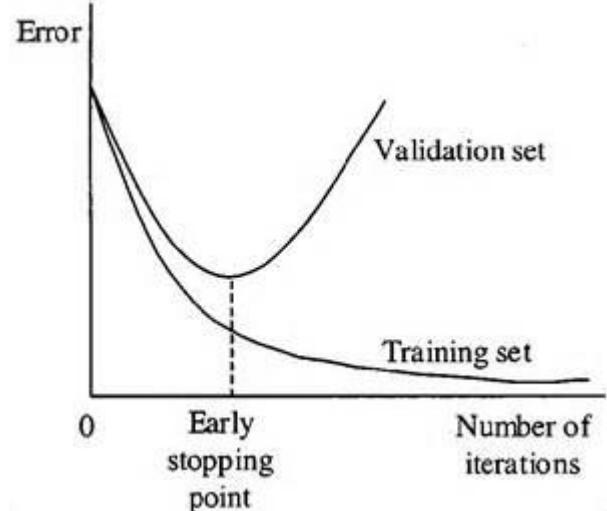
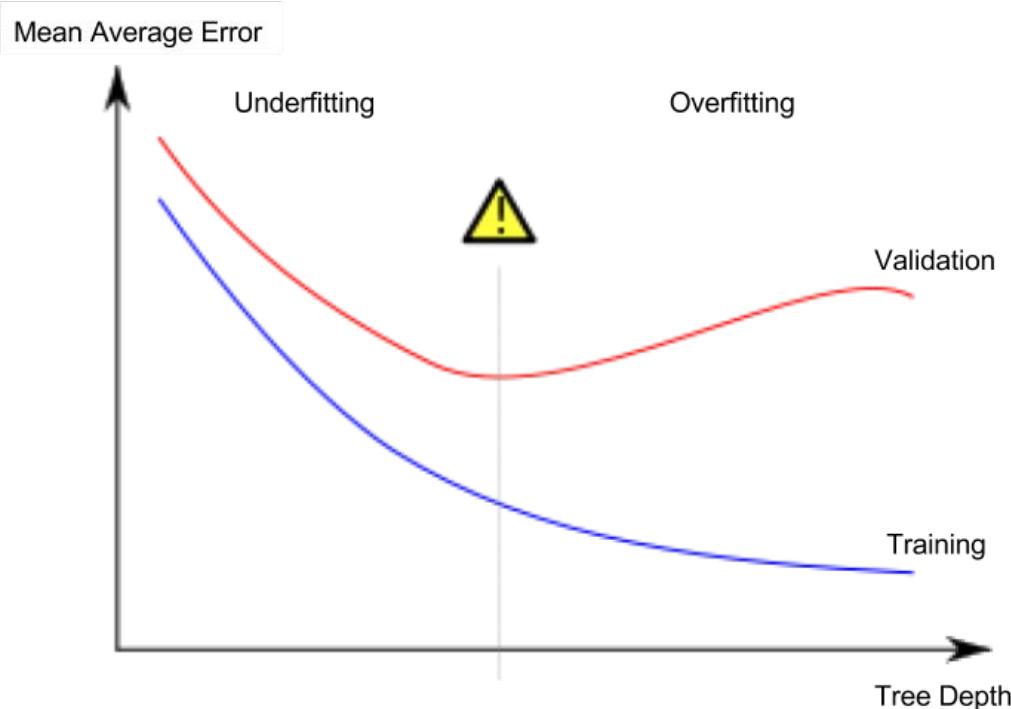
[ ]: # Load Data
df = pd.read_csv('../input/dataset/spam_or_ham.csv', delimiter=',', encoding='latin-1')
```

Below the code editor, there are tabs for "Console" and "Dataset". The "Dataset" tab shows a file named "spam_or_ham.csv" with dimensions 5572 x 2. There are icons for download, delete, and refresh.

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

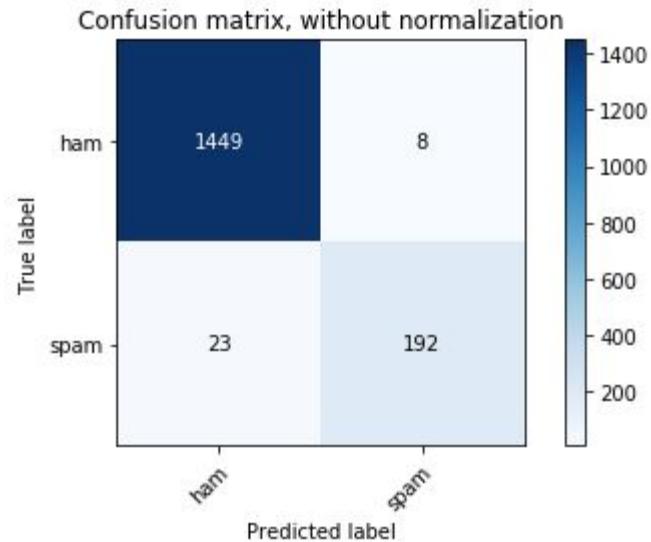
GROUPON[®]

Train Neural Network



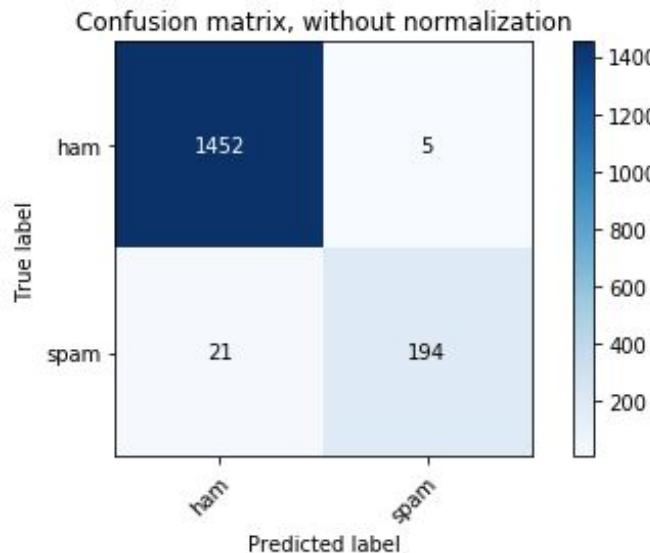
Avoid Overfit

Deep Learning vs Standard NLP



RNN

4s Epoch 3,
Accuracy: 0.980

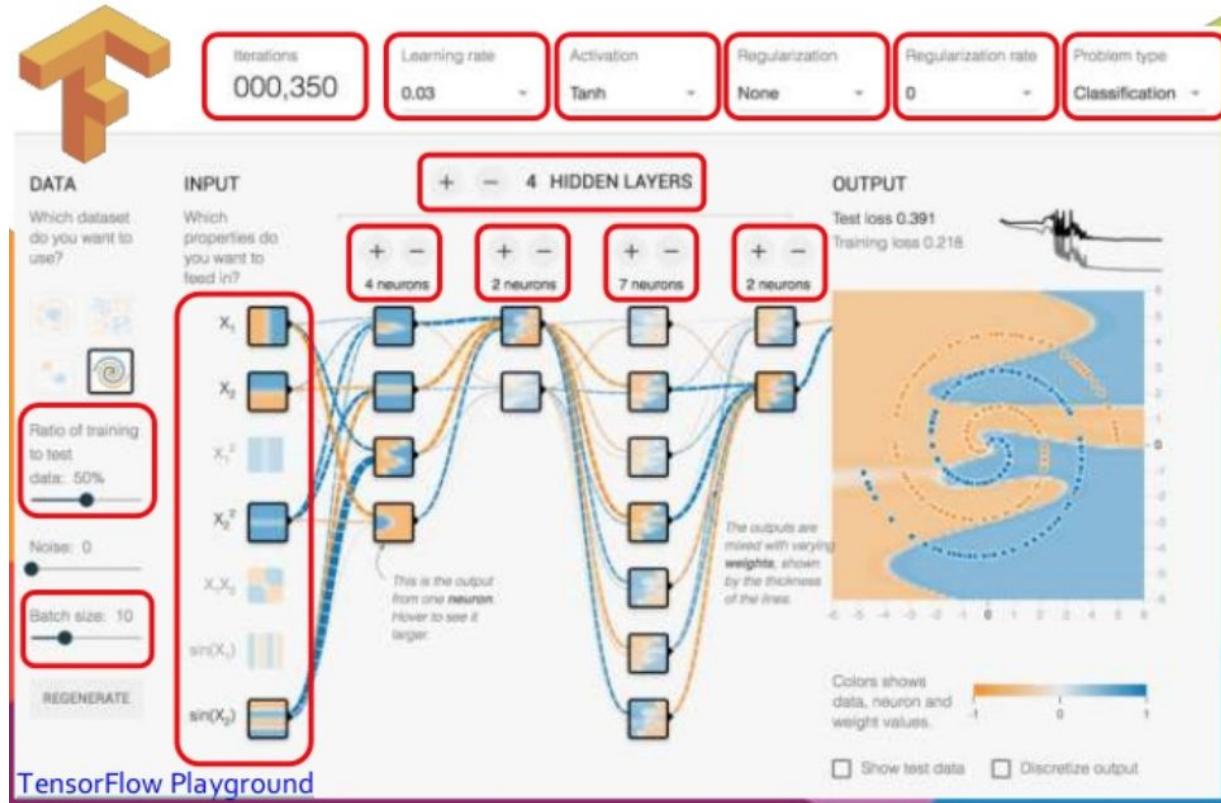


SVM

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

US Consumer Finance Complaints

US consumer complaints on financial products and company responses

Thousands of the consumer complaints weekly
about financial products and services

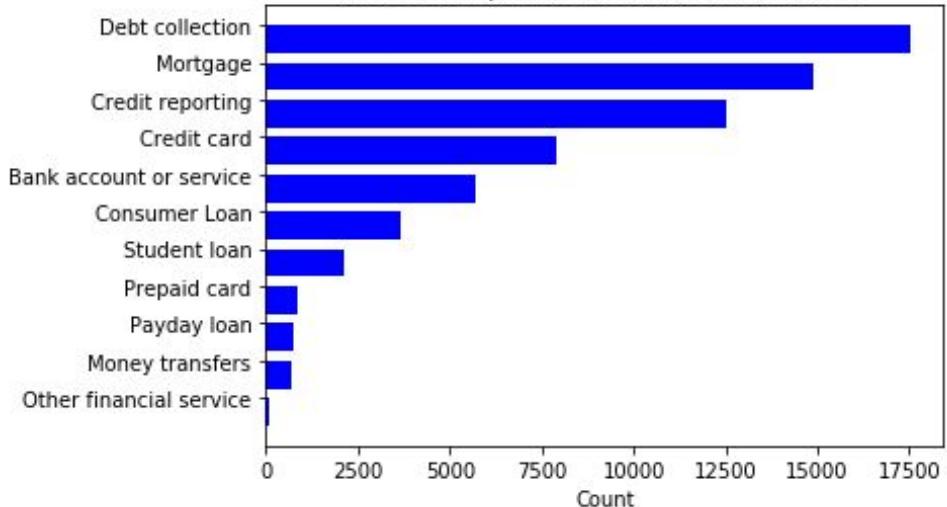
Need classification before sending final respond within 15 days

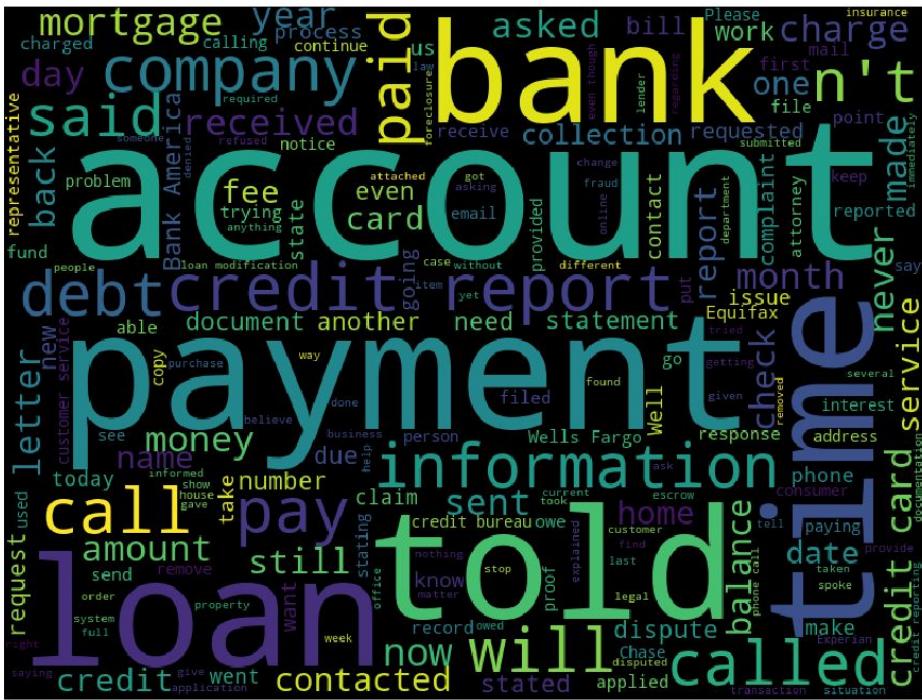
Randomly ~9% Acc

Data Train / Test

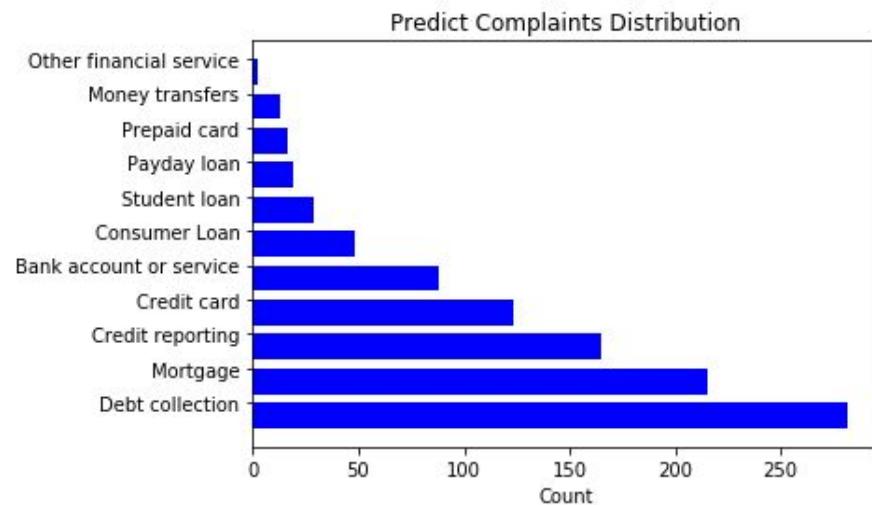
66.8k / 1k

Predict Complaints Distribution in Train Set





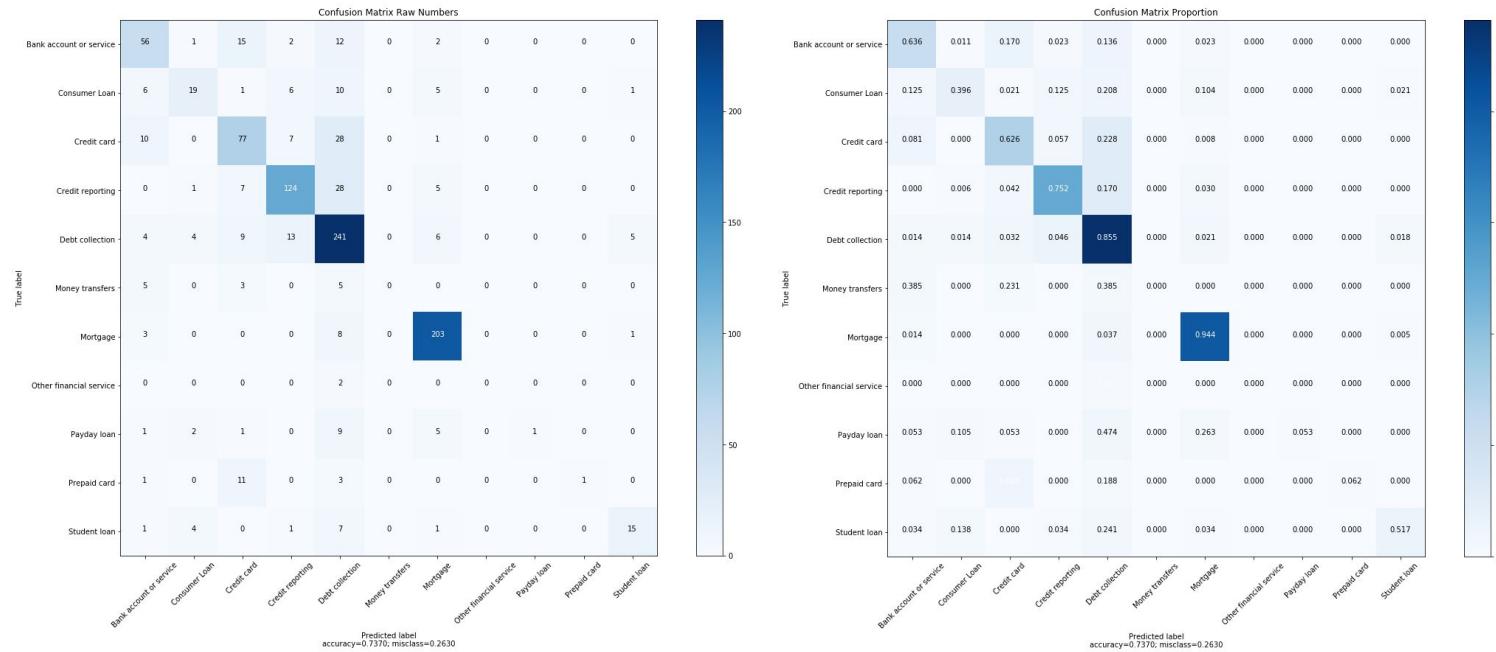
Test 1k Data



'Debt collection'

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

TensorFlow Model, train in 2 minutes



Final Prediction 73.7% on 1000 samples

GROUPON®

Transformation	Stemming	TF-IDF	Keras	DL Topology	Final Model
Data Filtering	Case Folding	Word2Vec	BRNN & LSTM		Ready to Deploy
Removing: - tags - duplicates	Removing: - Stop words - Double Letters	Data Enrichment (non Textual)	Naive Bayes Regression Linear SVM	Hyperparameters Grid Search Ensemble methods	Retrain and Update Model
Formatting	Add New Features	Other embeddings FastText ELMo	SVM (Kernel) Random Forests etc.	Confusion Matrix	

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

Deep Learning, model in 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

Github vs Standard vs Keras



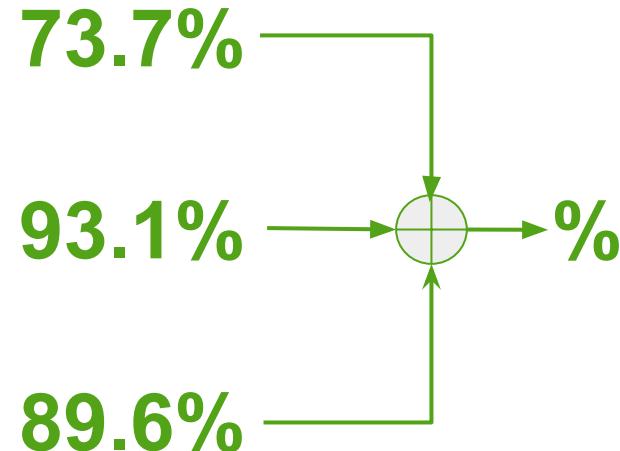
multi-class-text-classification-cnn

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

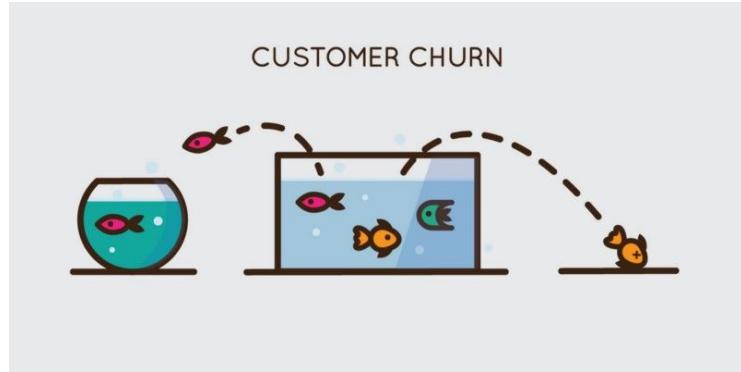


Keras RNN with BLSTM

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



Churn Modeling



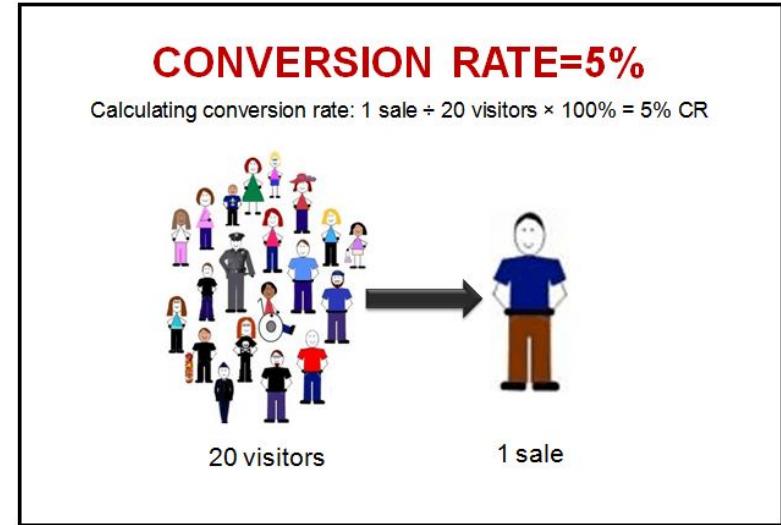
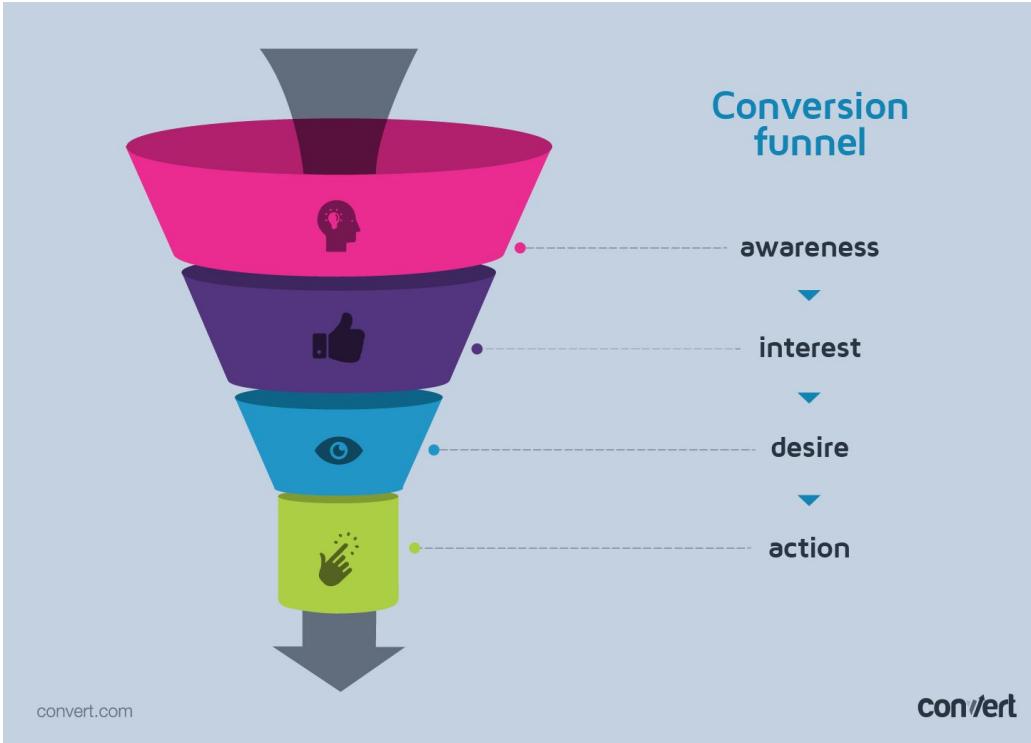
GROUPON®

SNA - Social Network Analysis

- ★ Leaders
- ★ Bridges

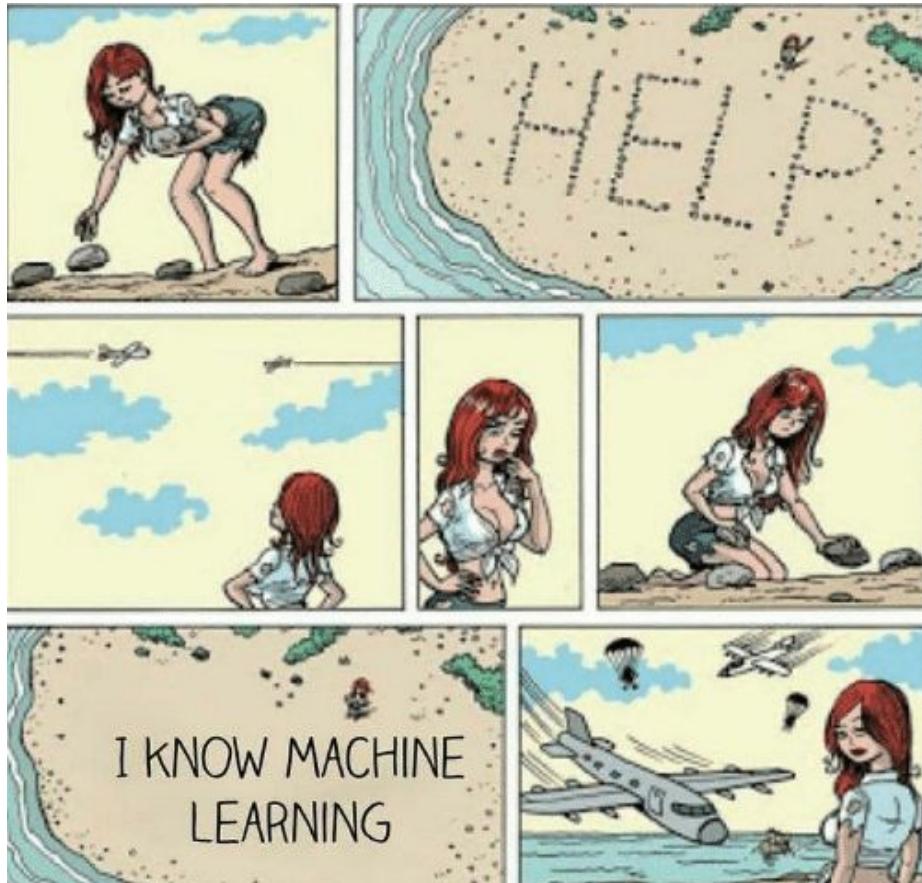


Conversion Funnel Analysis



<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