Deep learning in Fashion

Aditya Prasad Narisetty
Data Scientist

@adityaprasadn

- Machine Learning
- NeuralNets
- TF/Keras/PyTorch
- Decision Sciences

- Data Engineering
- Spark/Hive/ES/Cassandra
- Flask/Node/MQ/Redis

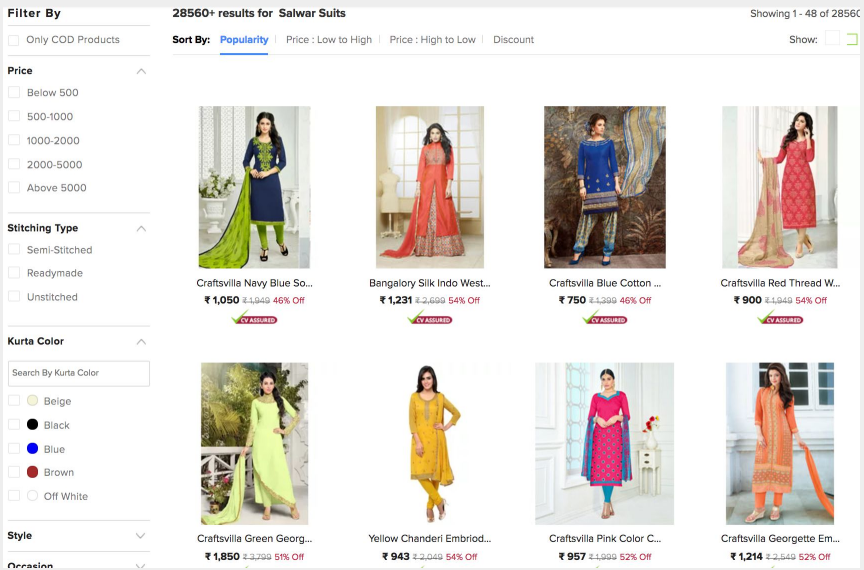
➢ Hiring @Dream11
➢ Craftsvilla/Housing/BwinParty

Fashion Industry

1. 400M Users, $20+bn e-retail
2. Huge Catalog, varied choices, startups
3. Market-place model
4. Huge competition, latest trends in fashion
5. Unverified products/images

Craftsvilla

1. India’s biggest online ethnic fashion
2. Miscategorized products
3. Duplicate/ripoffs
4. Limited color variants
5. Poor fashion visibility
6. Costly resizing for resolutions



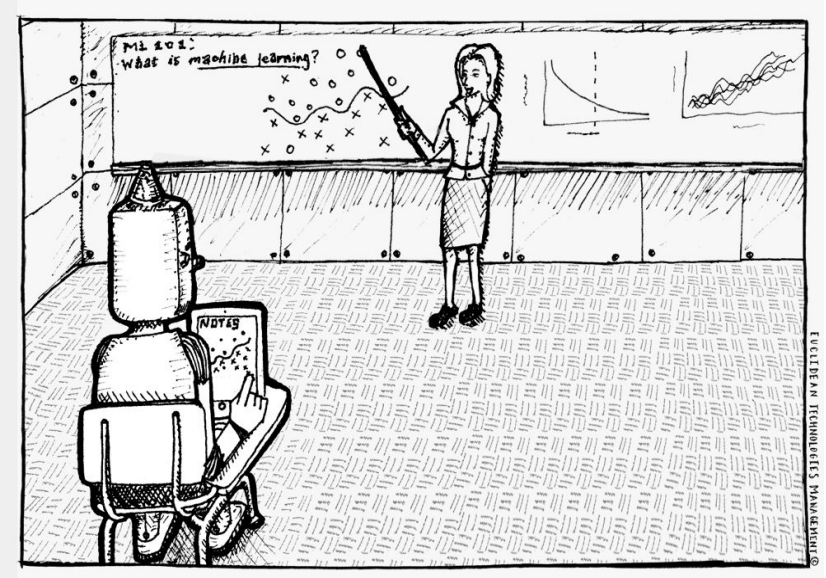
Why Automate?

The Pain

1. Slow
2. Costly
3. Offline

What If?

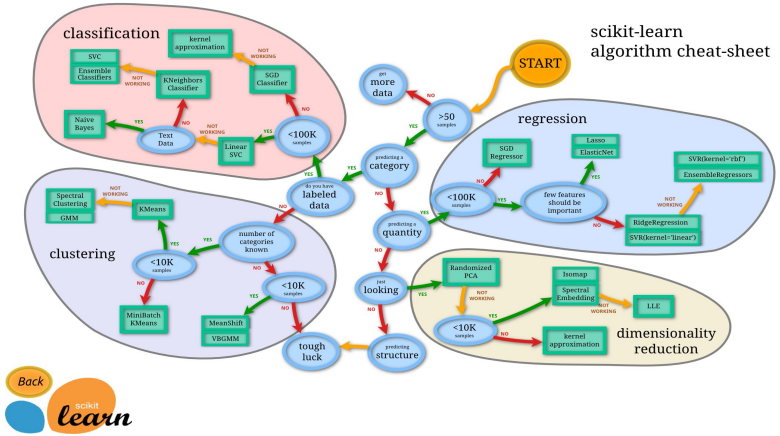
1. Classify Images into verticals
2. Identify abnormalities
3. Image to textual description
4. Scale it for real-time
5. Style it visually appealing/color variants



So it begins!

Project Eye:

- ✓ P-hash
- ✓ Duplicate detection
- ✓ 99.2% accuracy
- ✓ 1 use case



Brainstorm:

- ✓ CompVision. Border detection. Rule based engine.

  Huge investment in time, resources

- ✓ DCNNs for classification

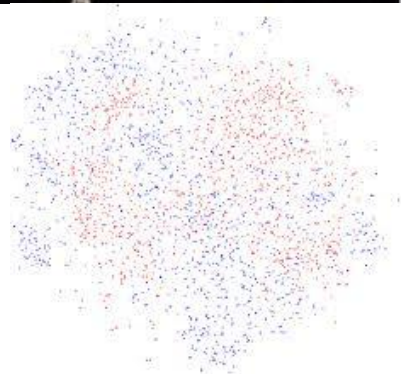
  low hanging fruit, state-of-the art performance

Model 1:

A small DCNN to classify images into their respective categories

goo.gl/zlz36Y

- Model trained from raw 32x32 images
- Weights learnt from scratch
- 4 Hidden Layers, Maxpool, relu, 2 fully connected layers, 0.5 dropout
- 50k training data(2:1 train-test split)
- 3 class classification with 64.5% accuracy. 2X better to random model
- TensorFlow, categorical_cross_entropy, adam_opt



Error Analytics:

1. T-SNE visualization of final hidden layer
2. Images with similar faces, borders, colors were learnt :(
3. Tensorboard to the rescue!

Meet the evolution

- 150k of 64x64x3 images
- Cleaned up images
- TF transfer learning - InceptionV3
- 3-class classification. 96.4% accuracy. All hail Ultron!
- 96.53% accuracy when faces-masked
- 8-class classification. 92.3%

1. Model Struggled to differentiate similar fashions. Insufficient information? Resolution, learning capacity,
2. Misclassified multi-object, huge background noise
3. Huge training time. 15 days on g2.2x large GPU instance. Scale?!

5 LoC: goo.gl/LMdt2A


$$Cost?!$$
NVIDIA
TESLA

Production Rollout

- 500k images
  a. 4*g2.2/ g2.4 xlarge
  b. 1000+ hours of training on g2.2xlarge
  c. $3000 to train for a classifier.

- How about training on 2M images?
  a. For 8-class classifier?
  b. For multi-label hierarchical classification

Solution:

a. Spot instances from Brazil, Tokyo, Sydney
b. Check for lowest prices, spin up
c. Tensorflow train
d. Checkpoints persisted to S3
e. Cron, docker deploy, pull, train, persist, update!
f. 25 days of Deep Eye: $120

What lies ahead?

- Update mislabled SKUs
- Add more hidden layers for further generalization
- final layer of DNN --- embedded space for images --- style-transfer(GANs and Auto-encoders)
- Multi-label hierarchical classification to generate website-breadcrumb.
- Generate Product description from output layer.

  Image vectors::tags::LSTM

Start here

- Scrap a few images of Indian sweets. 20/category
- Organize training images in separate category folders
- Train with CNNs from scratch
- Train with transfer learning
- Visualize model learning with Tensorboard/T-SNE
- Explore GoogleCloudVision APIs
- FloydHub for cost effective faster and iterative model development
- Explore Pytorch

Thank You

@adityaprasadn