Microsoft

Labeling Data on a Budget with Active Learning and Transfer Learning

Mario Inchiosa
Bob Horton
Ali Zaidi
Omar Alonso

Provisioning your Data Science VM

- Free Microsoft Azure accounts with $200 credit
- Windows or Linux
- Provision single Data Science VMs manually or many via automation

- For instructions, see README here:

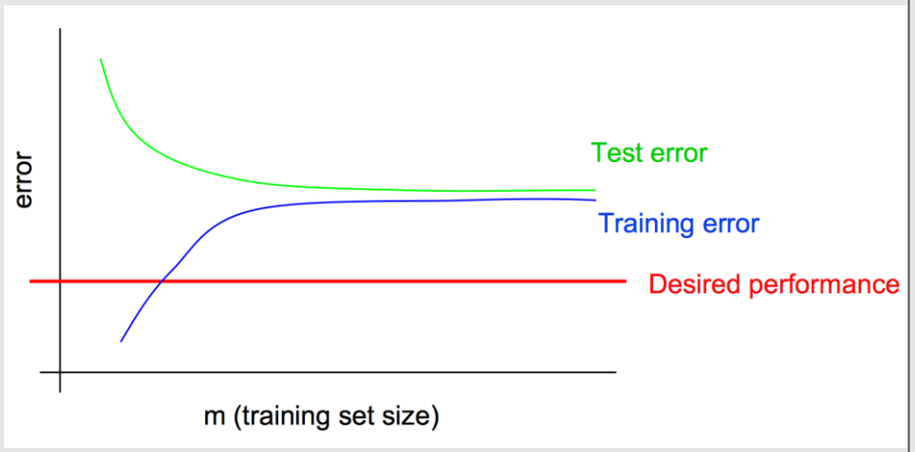
https://github.com/Azure/active-learning-workshop

I have plenty of data, but getting it labeled is expensive

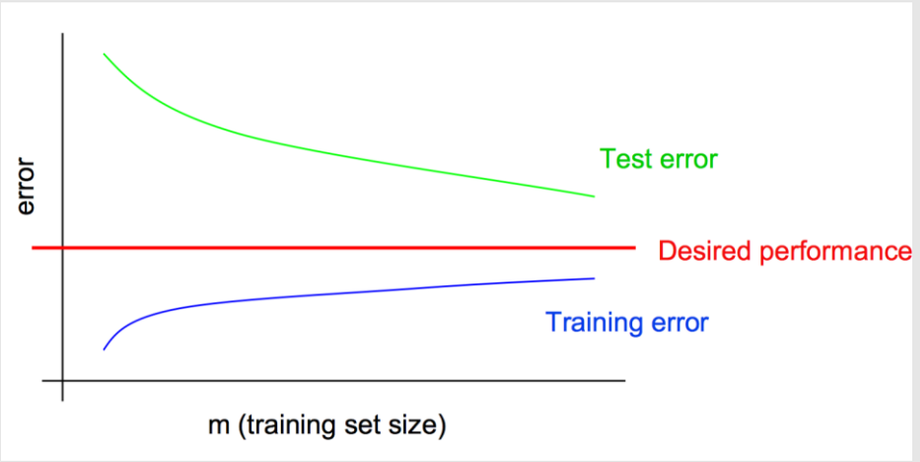
- **pre-trained deep learning models** and **word embeddings** let you generate features that can be used in traditional machine learning approaches.
- these features let you use **low-complexity models** that can learn from small numbers of cases.
- **active learning** lets you take advantage of large sets of unlabeled data to build more accurate classifiers by selecting the most useful additional examples to label for training.

Bias-Variance Tradeoff and Performance Diagnostics

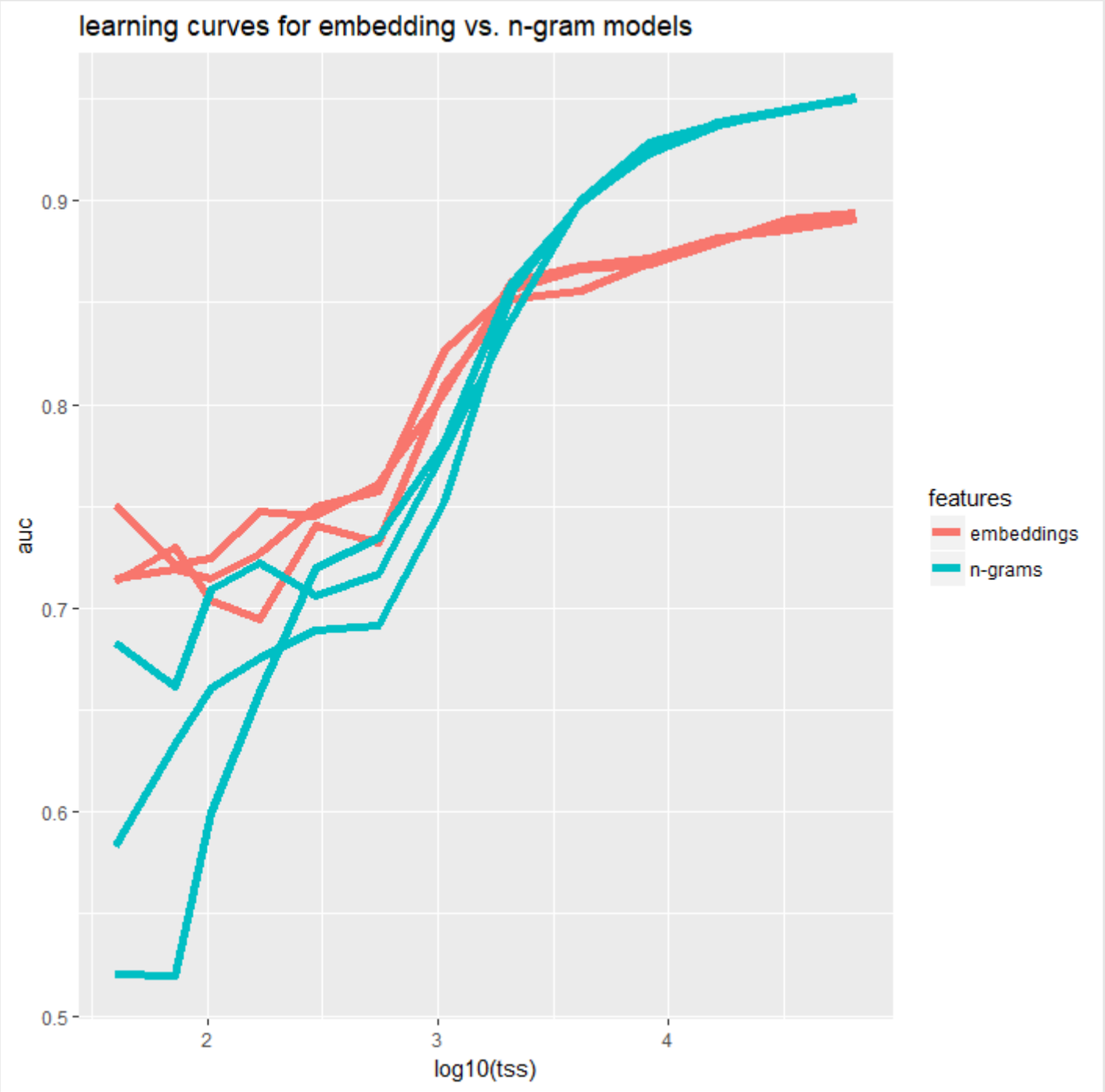
- We have an intrinsic need to balance the capacity of our models with the amount of data we have
- What turns out to be especially important, is the amount of labeled data that we can turn into features



- Use better features (differentiable programming to automatically learn good features)
- Use a higher capacity model



- Label more data
- Apply regularization / shrinkage


learning curves for embedding vs. n-gram models
auc
0.9
0.8
0.7
0.6
0.5
2
3
4
log10(tss)
features
embeddings
n-grams

Use Cases

“Wiki detox”

- Active learning from text data.
- Binary classifier: is this comment a personal attack?
- Featurization from pre-trained word embeddings.

Wood knot images

- Active learning from image data.
- Multi-class classifier: which type of knot is this
- Featurization from pre-trained deep learning model (Resnet)

Featurizing Text

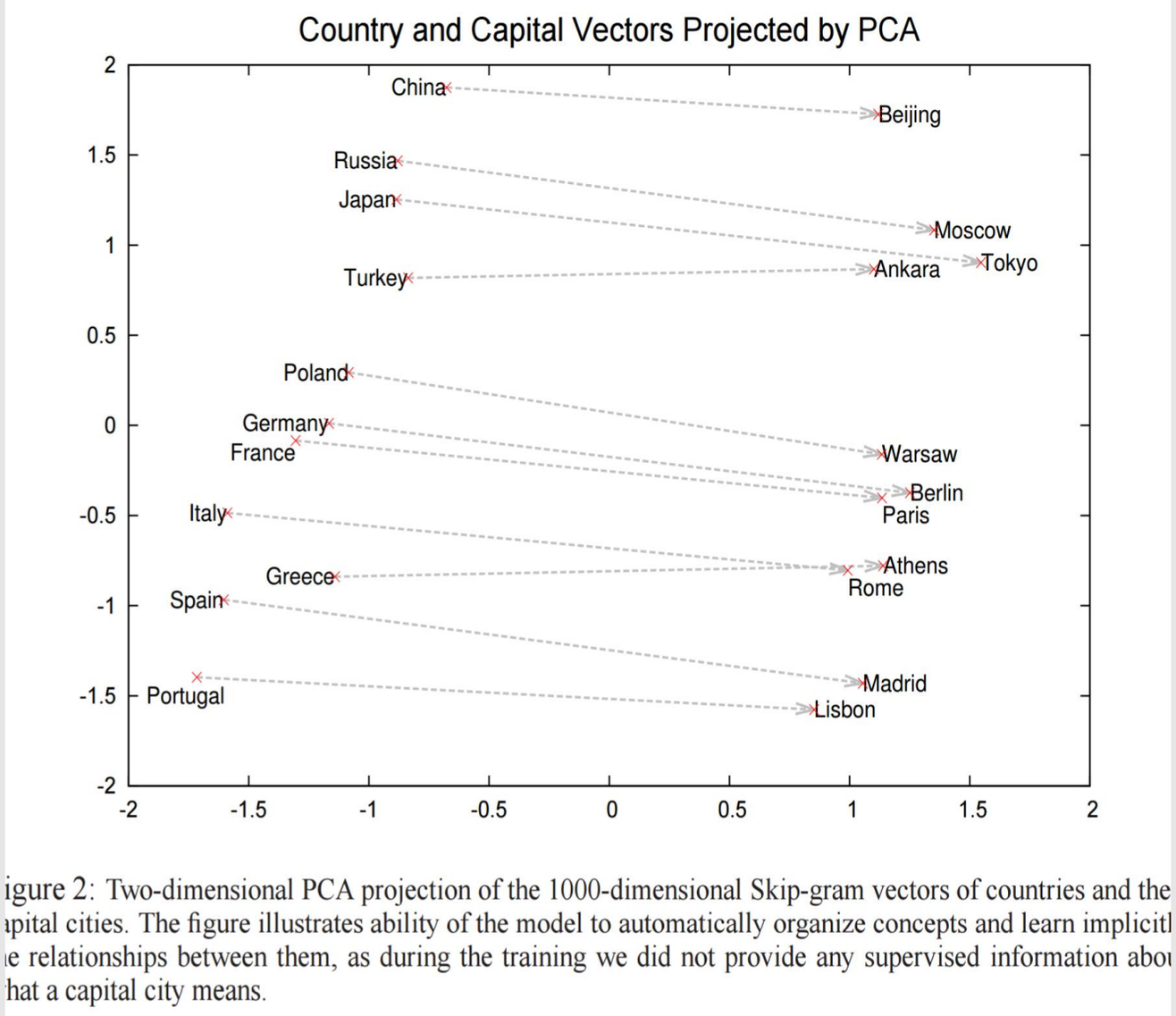
data, data everywhere

Word analogies with word2vec

[king] – [man] + [woman] ≈ [queen]

Table 1: *Examples of five types of semantic and nine types of syntactic questions in the Semantic-Syntactic Word Relationship test set.*

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks



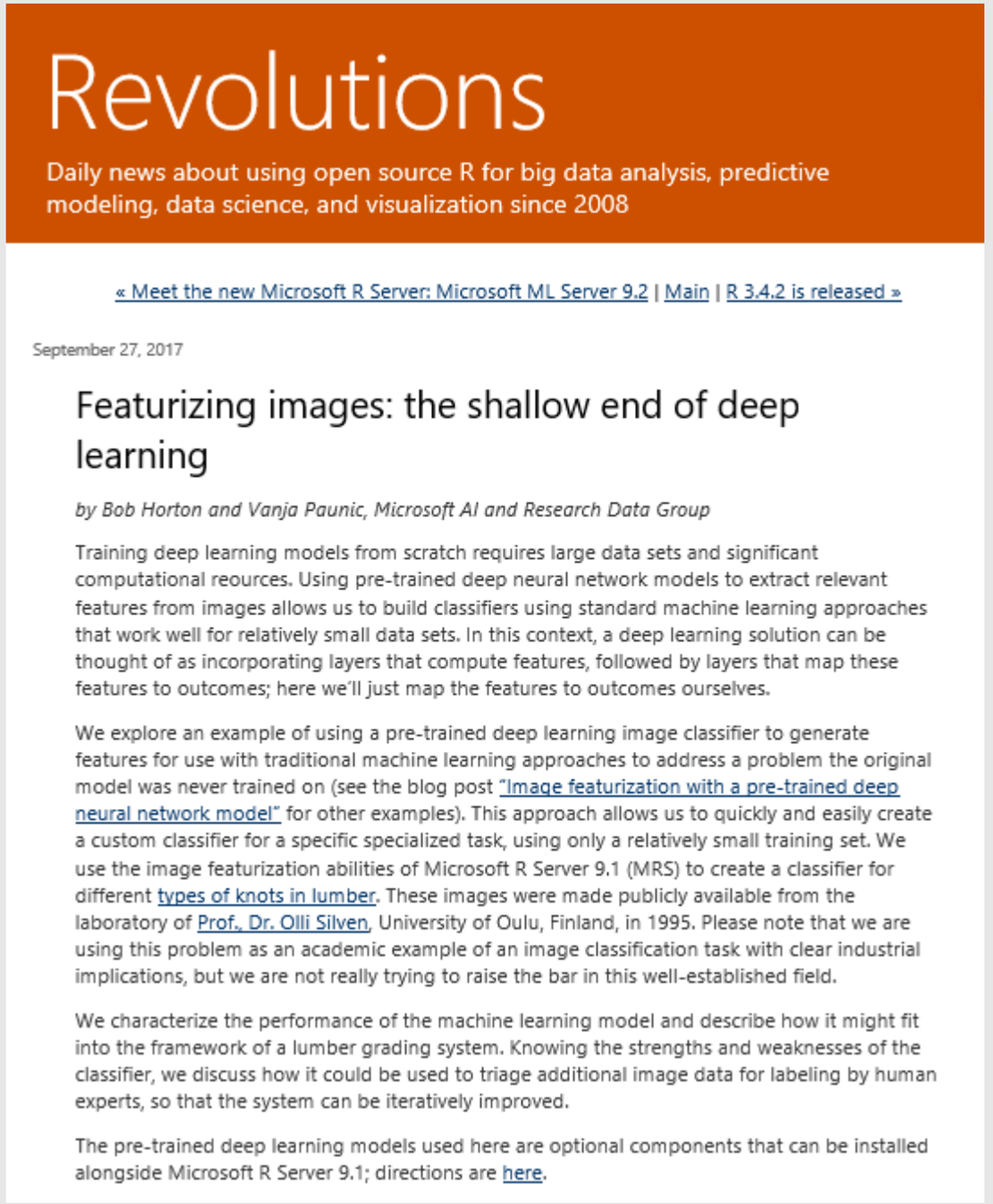
igure 2: Two-dimensional PCA projection of the 1000-dimensional Skip-gram vectors of countries and the apital cities. The figure illustrates ability of the model to automatically organize concepts and learn implicitl e relationships between them, as during the training we did not provide any supervised information abou hat a capital city means.

(Mikolov et al., 2013)

Microsoft

Featurizing images: the shallow end of deep learning

- http://blog.revolutionanalytics.com/2017/09/wood-knots.html



Revolutions

Daily news about using open source R for big data analysis, predictive modeling, data science, and visualization since 2008

« Meet the new Microsoft R Server: Microsoft ML Server 9.2 | Main | R 3.4.2 is released »

September 27, 2017

Featurizing images: the shallow end of deep learning

by Bob Horton and Vanja Paunic, Microsoft AI and Research Data Group

Training deep learning models from scratch requires large data sets and significant computational reources. Using pre-trained deep neural network models to extract relevant features from images allows us to build classifiers using standard machine learning approaches that work well for relatively small data sets. In this context, a deep learning solution can be thought of as incorporating layers that compute features, followed by layers that map these features to outcomes; here we'll just map the features to outcomes ourselves.

We explore an example of using a pre-trained deep learning image classifier to generate features for use with traditional machine learning approaches to address a problem the original model was never trained on (see the blog post "Image featurization with a pre-trained deep neural network model" for other examples). This approach allows us to quickly and easily create a custom classifier for a specific specialized task, using only a relatively small training set. We use the image featurization abilities of Microsoft R Server 9.1 (MRS) to create a classifier for different types of knots in lumber. These images were made publicly available from the laboratory of Prof., Dr. Olli Silven, University of Oulu, Finland, in 1995. Please note that we are using this problem as an academic example of an image classification task with clear industrial implications, but we are not really trying to raise the bar in this well-established field.

We characterize the performance of the machine learning model and describe how it might fit into the framework of a lumber grading system. Knowing the strengths and weaknesses of the classifier, we discuss how it could be used to triage additional image data for labeling by human experts, so that the system can be iteratively improved.

The pre-trained deep learning models used here are optional components that can be installed alongside Microsoft R Server 9.1; directions are here.

Types of wood knots

Sound knot: A knot grown firmly into the surrounding wood material and does not contain any bark or signs of decay. The color may be very close to the color of sound wood.

Dry knot: A firm or partially firm knot, and has not taken part to the vital processes of growing wood, and does not contain any bark or signs of decay. The color is usually darker than the color of sound wood, and a thin dark ring or a partial ring surrounds the knot.

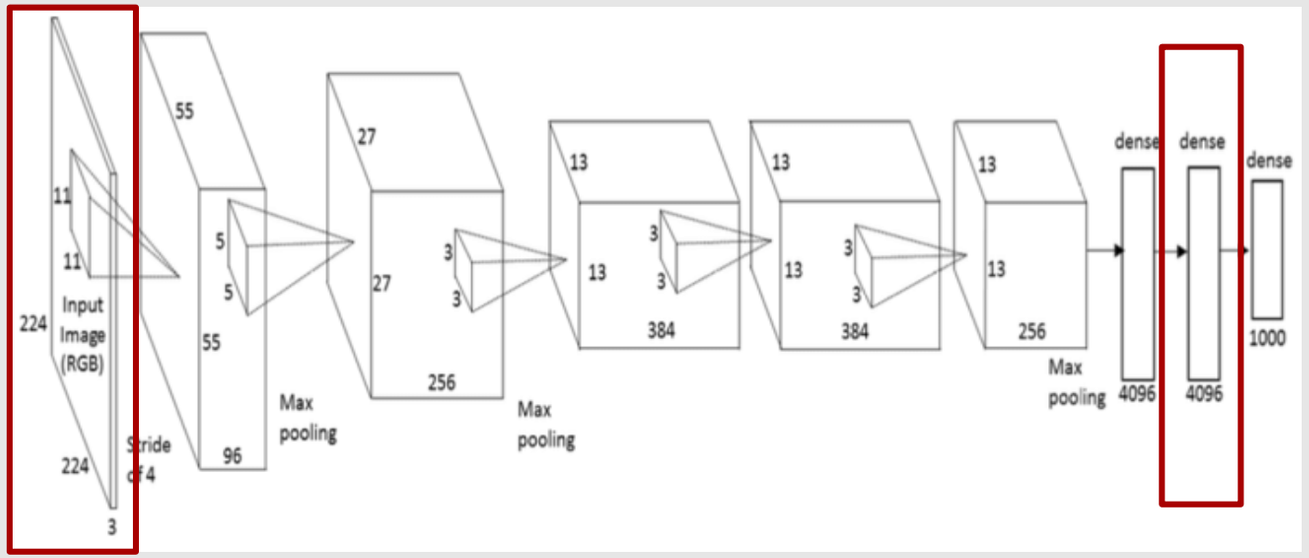
Encased knot: A knot surrounded totally or partially by a bark ring. Compared to dry knot, the ring around the knot is thicker.

Image Featurization in Microsoft ML Server

- Starting with Microsoft R Server 9.1, the MicrosoftML package has added support for pre-trained deep neural network models for image featurization.

- We can now use the following four deep neural network models trained on ImageNet data set to extract features from images.
  - ResNet-18
  - ResNet-50
  - ResNet-101
  - AlexNet

Input image



Output features

Image Featurization in Microsoft ML Server

- Usage of these pre-trained models allows us to take advantage of their features hard learned from previous data sets which would be otherwise impossible or very inefficient to feature engineer.

- Heuristically, the larger the model, the better the performance but the longer it takes to run.

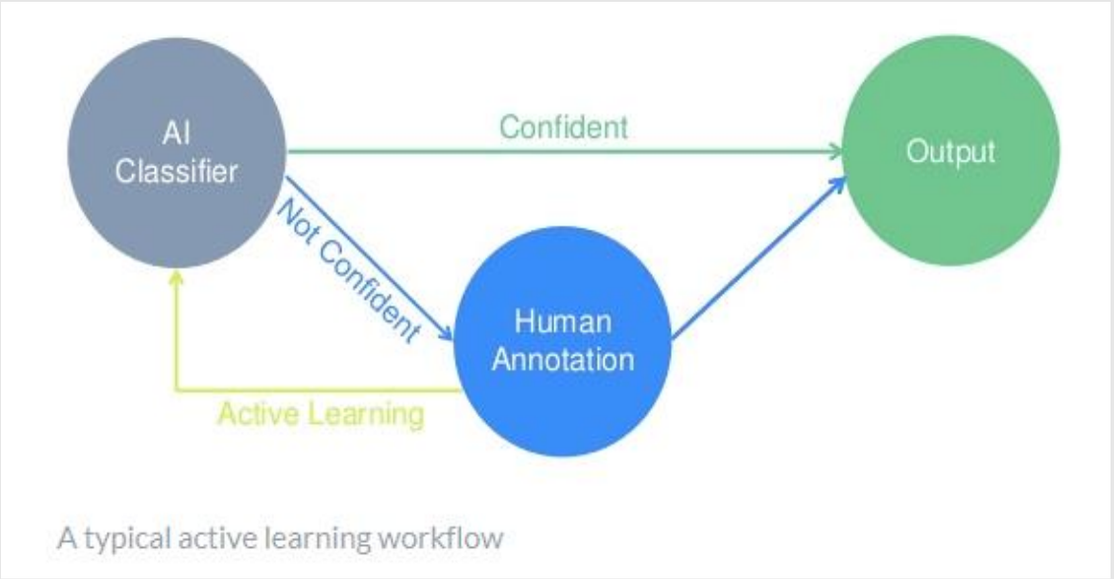
```
#define tranform and train model
mlTransform=list(loadImage(vars = list(Image = "Path")),
                 resizeImage(vars = "Image", width = 224, height = 224, resizingOption = "IsoPad"),
                 extractPixels(vars = list(Pixels = "Image")),
                 featurizeImage(var = "Pixels", outVar = "Feature", dnnModel = "resnet101"))

model <- rxLogisticRegression(Label ~ Feature, data = train_df,
          mlTransforms = mlTransform, mlTransformVars = "Path")
```

* Code snapshot taken from https://blogs.msdn.microsoft.com/mlserver/2017/04/12/image-featurization-with-a-pre-trained-deep-neural-network-model/

Active Learning

- Active learning is a case of semi-supervised learning in which an algorithm interactively asks for additional labeled data that would result in most gain in model performance
- Data (unlabeled) is often easier to come by than expert labelers
- Active learning starts with a preliminary classifier and looks for the samples that it has the most to learn from
  - What is the model good at? What needs work (e.g., more training data)?
  - How much of the unlabeled data can we eliminate as already identifiable?



A typical active learning workflow

* Image taken from https://www.crowdflower.com

Query strategies[edit]

Algorithms for determining which data points should be labeled can be organized into a number of different categories:[1]

- **Uncertainty sampling:** label those points for which the current model is least certain as to what the correct output should be
- **Query by committee:** a variety of models are trained on the current labeled data, and vote on the output for unlabeled data; label those points for which the "committee" disagrees the most
- **Expected model change:** label those points that would most change the current model
- **Expected error reduction:** label those points that would most reduce the model's generalization error
- **Variance reduction:** label those points that would minimize output variance, which is one of the components of error
- **Balance exploration and exploitation:** the choice of examples to label is seen as a dilemma between the exploration and the exploitation over the data space representation. This strategy manages this compromise by modelling the active learning problem as a contextual bandit problem. For example, Bouneffouf et al.[7] propose a sequential algorithm named Active Thompson Sampling (ATS), which, in each round, assigns a sampling distribution on the pool, samples one point from this distribution, and queries the oracle for this sample point label.
- **Exponentiated Gradient Exploration for Active Learning:**[8] In this paper, the author proposes a sequential algorithm named exponentiated gradient (EG)-active that can improve any active learning algorithm by an optimal random exploration.

Wikipedia: Active learning (machine learning)
https://en.wikipedia.org/w/index.php?title=Active_learning_(machine_learning)

Label Quality

Careful with the data

In the era of big data and machine learning

labels -> features -> predictive model -> optimization

Labeling/experimentation perceived as boring

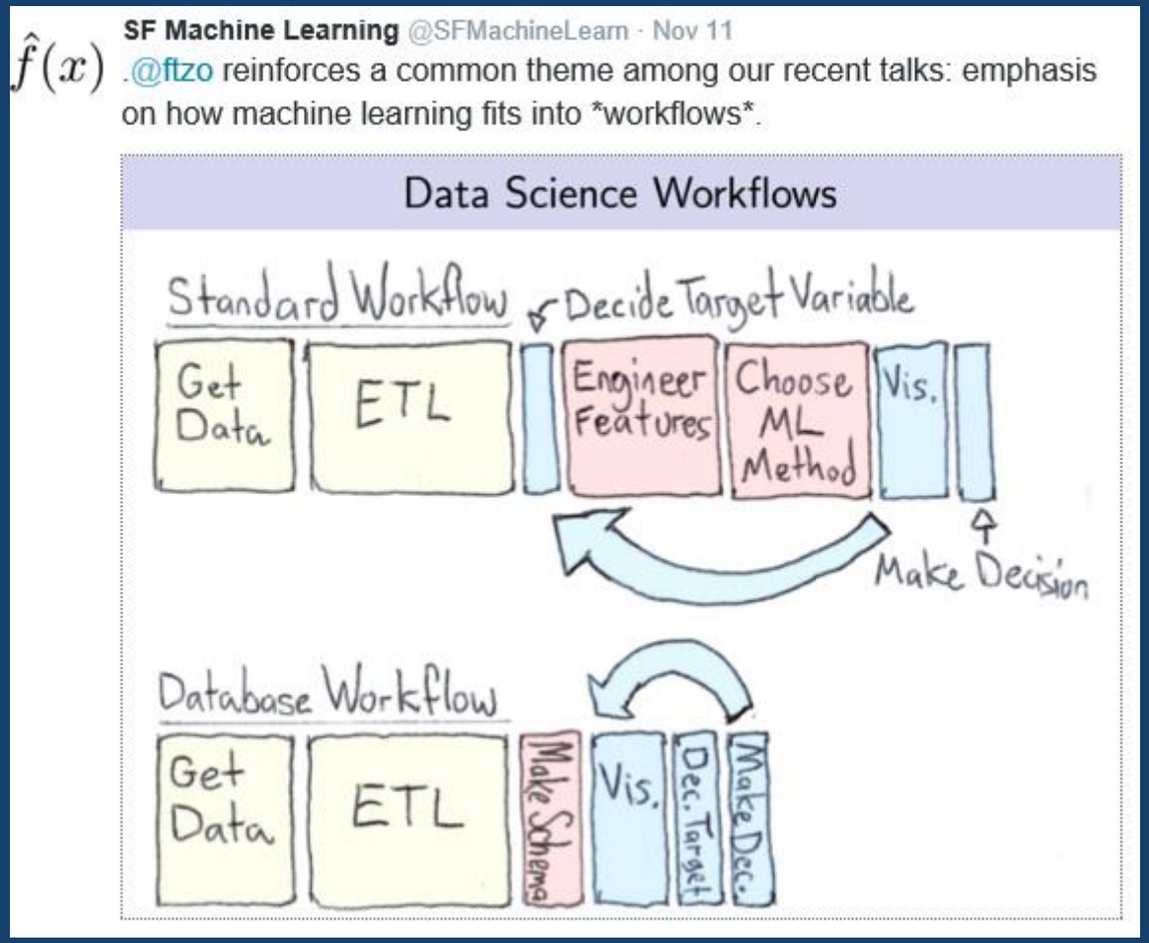
Tendency to rush labeling

Label quality is very important

Garbage in, garbage out

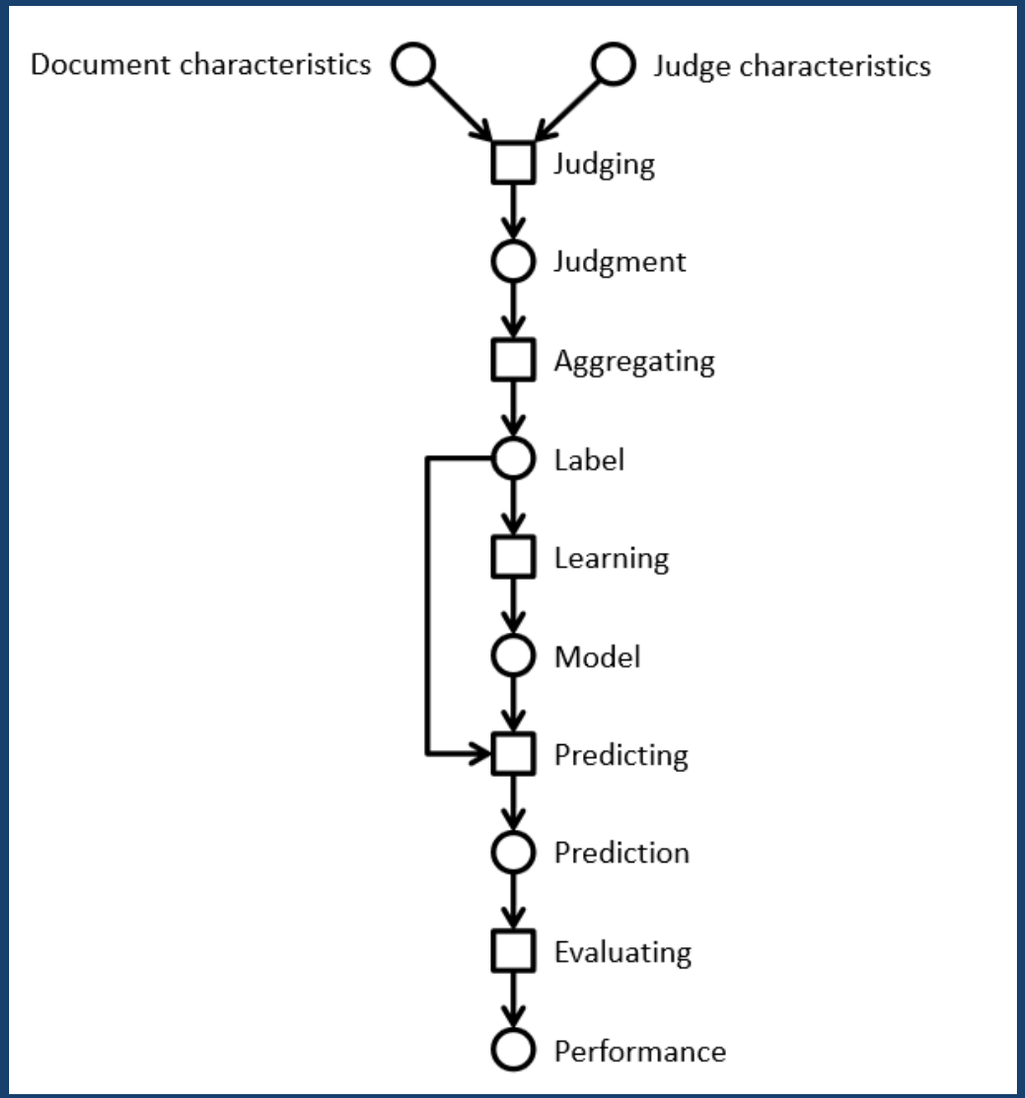
Big data, ML, and data science

Labels -> features -> predictive model -> optimization



Lifecycle of a label

Information retrieval example



Using a crowd to label a data set

Using ML to process the complete data set

Label quality

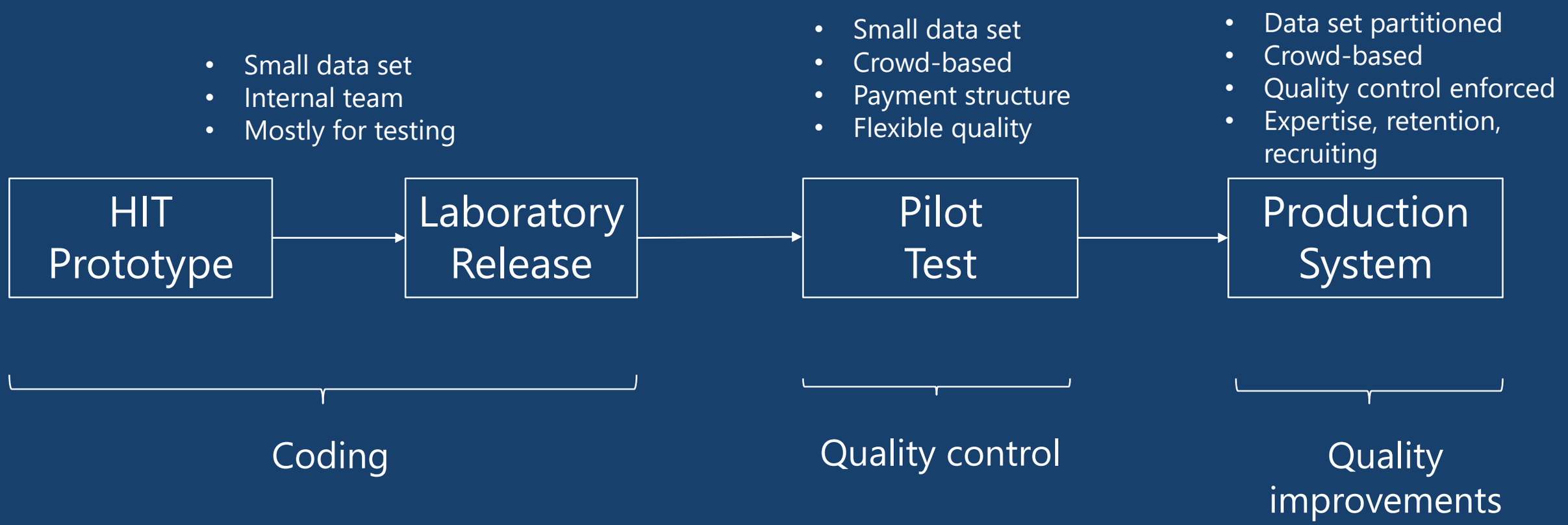
Quality

Meets internal customer needs
Free from deficiencies

Process

Don’t rush labeling
Don’t outsource
Own it end to end
Large scale
Continuity

Development process



Microsoft