

Analytics

With spark and mllib

Content pre lunch

1. Intro to Analytics

- a. What is analytics (machine learning)
- b. When can it be useful
- c. Different kinds of machine learning
- d. Different applications of machine learning

2. Some Common algorithms

- a. Linear and Logistic Regression
- b. Artificial Neural Networks
- c. Tree based
- d. KMeans
- e. Collaborative Filtering

3. ML Tools and Spark mllib

Content post lunch

Individual hands-on exercises

What is ~~analytics~~?

Big Data

What is Big Data?

40 ZETTABYTES

[43 TRILLION GIGABYTES]
of data will be created by
2020, an increase of 300
times from 2005

6 BILLION
PEOPLE
have cell
phones

WORLD POPULATION: 7 BILLION

Volume
SCALE OF DATA

It's estimated that
2.5 QUINTILLION BYTES
[2.3 TRILLION GIGABYTES]
of data are created each day

Most companies in the
U.S. have at least
100 TERABYTES
[100,000 GIGABYTES]
of data stored

The New York Stock Exchange
captures

1 TB OF TRADE
INFORMATION
during each trading session



Velocity
ANALYSIS OF
STREAMING DATA

By 2016, it is projected
there will be

18.9 BILLION
NETWORK
CONNECTIONS

— almost 2.5 connections
per person on earth



Modern cars have close to
100 SENSORS
that monitor items such as
fuel level and tire pressure



The FOUR V's of Big Data

From traffic patterns and music downloads to web history and medical records, data is recorded, stored, and analyzed to enable the technology and services that the world relies on every day. But what exactly is big data, and how can these massive amounts of data be used?

As a leader in the sector, IBM data scientists break big data into four dimensions: **Volume, Velocity, Variety and Veracity**

Depending on the industry and organization, big data encompasses information from multiple internal and external sources such as transactions, social media, enterprise content, sensors and mobile devices. Companies can leverage data to adapt their products and services to better meet customer needs, optimize operations and infrastructure, and find new sources of revenue.

By 2015
4.4 MILLION IT JOBS
will be created globally to support big data,
with 1.9 million in the United States



As of 2011, the global size of
data in healthcare was
estimated to be
150 EXABYTES
[161 BILLION GIGABYTES]



30 BILLION
PIECES OF CONTENT
are shared on Facebook
every month



Variety
DIFFERENT
FORMS OF DATA



By 2014, it's anticipated
there will be
420 MILLION
WEARABLE, WIRELESS
HEALTH MONITORS

4 BILLION+
HOURS OF VIDEO
are watched on
YouTube each month



400 MILLION TWEETS
are sent per day by about 200
million monthly active users

1 IN 3 BUSINESS
LEADERS

don't trust the information
they use to make decisions



27% OF
RESPONDENTS

In one survey were unsure of
how much of their data was
inaccurate

Veracity
UNCERTAINTY
OF DATA

Poor data quality costs the US
economy around

\$3.1 TRILLION A YEAR



Big Data

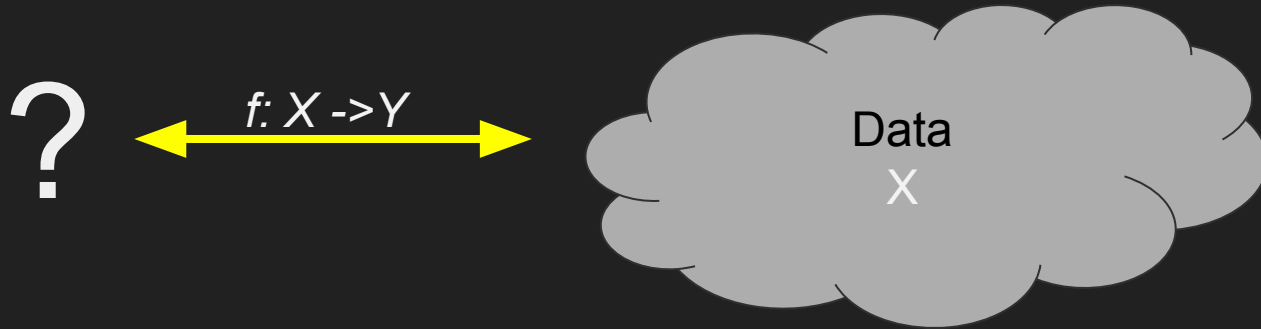
- Data Storage
- Data Processing
- Visualization
- Analytics

What is analytics?

aka

- Predictive Analytics (Business)
- **Machine Learning** (CS)
- Statistical Learning (Statistics)
- etc

What is Analytics?



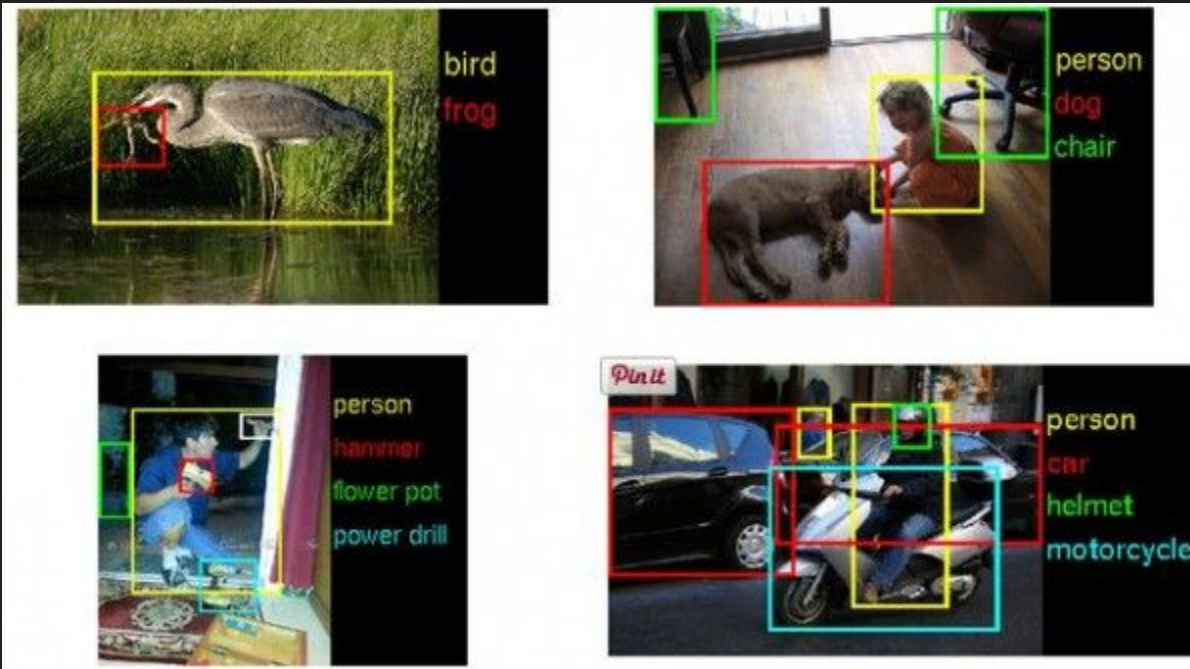
$$p(v) = \frac{4v^2}{\sqrt{\pi}} \left(\frac{m}{2kT} \right)^{3/2} e^{-\frac{mv^2}{2kT}}$$

When can it be useful

Physical model not available or too complex/expensive to construct, e.g. ?="What's in the image"



When can it be useful



Different kinds of Machine Learning

Supervised - “training model **with** a teacher”. $f(\mathbf{x})=\mathbf{y}$. Your data $D=\{(\mathbf{x},\mathbf{y})\}$

Unsupervised - “training model **without** a teacher”. $f(\mathbf{x})$. Your data $D=\{\mathbf{x}\}$.

Reinforcement Learning - Feedback from actions are used to train the model.

Applications (Supervised)

Regression

- How much (mm) will it rain tomorrow?
- How many t-shirts will we sell next week?
- How much energy will this wind power plant produce tomorrow, given this conditions?

Classification

- Will it rain tomorrow?
- Which size of t-shirts will sell most next week?
- Is the weather conditions good enough for the wind power plant to be on tomorrow?

Application (Unsupervised)

Clustering

- Who likes the same kind of movies (social network analysis)?
- Group homologous sequences into gene families.
- Customer Segmentation.

Anomaly Detection

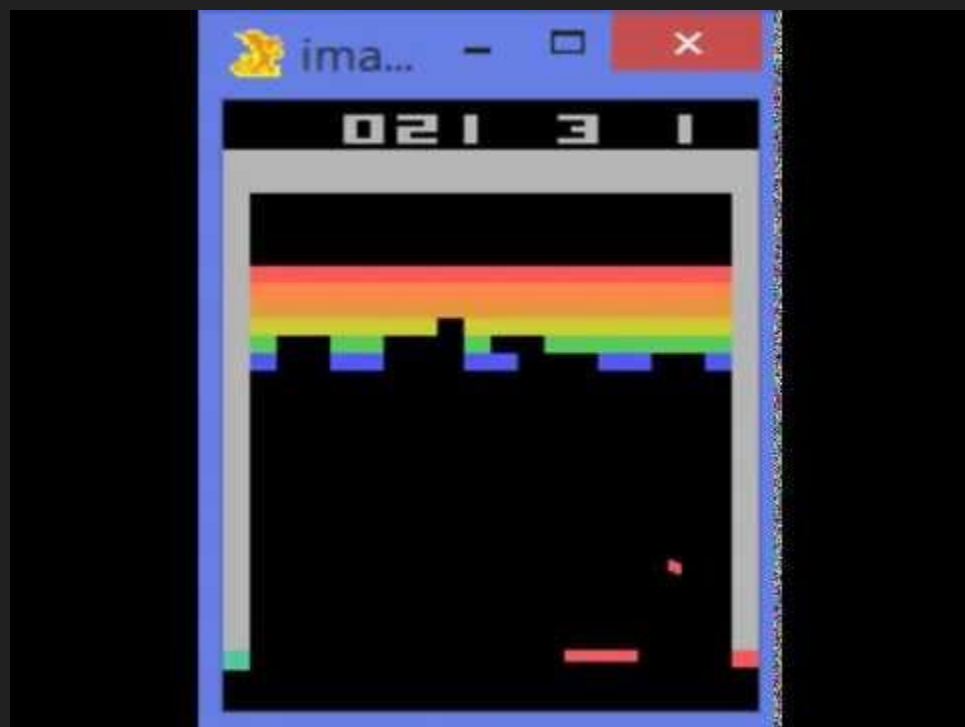
- Is this bearing faulty based on its vibrations?
- Is this a fraudulent use of this credit card?

Recommender Systems

- This user likes movies x, y and z, which movie is he/she most likely to like besides those?

Application (Reinforcement)

- Computer playing games
- Autonomous driving

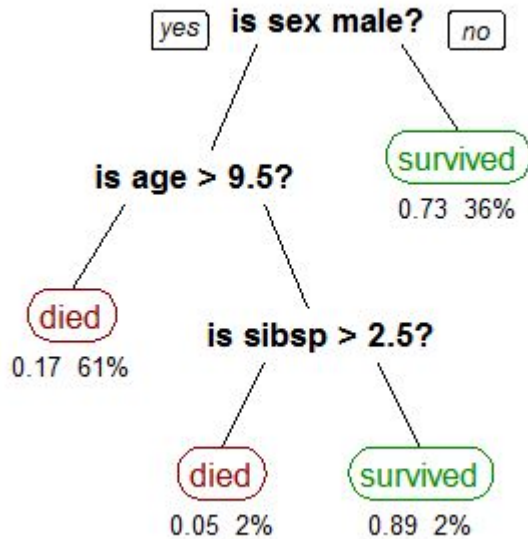




2. Some ML Algorithms

Linear Reg. & Logistic Reg. & Neural Networks

Tree based models (CART)



A tree showing survival of passengers on the [Titanic](#)

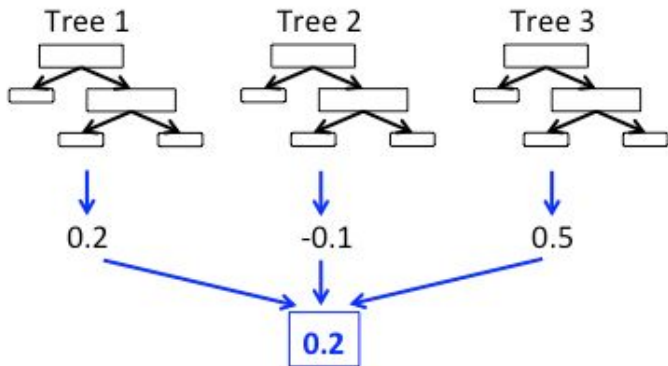
Tricks with trees (Ensemble and Boosting)

Random Forest

70 % data

70 % variables

Ensemble Model:
example for regression



Gradient Boosting (Simple Version)

(Why is it called “gradient”?)

(Answer next slides.)

(For Regression Only)

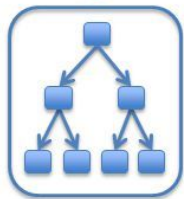
$$S = \{(x_i, y_i)\}_{i=1}^N$$

$$h(x) = h_1(x) + h_2(x) + \dots + h_n(x)$$

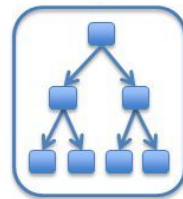
$$S_1 = \{(x_i, y_i)\}_{i=1}^N$$

$$S_2 = \{(x_i, y_i - h_1(x_i))\}_{i=1}^N$$

$$S_n = \{(x_i, y_i - h_{1:n-1}(x_i))\}_{i=1}^N$$

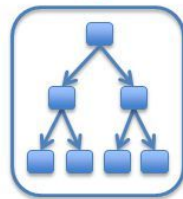


$h_1(x)$



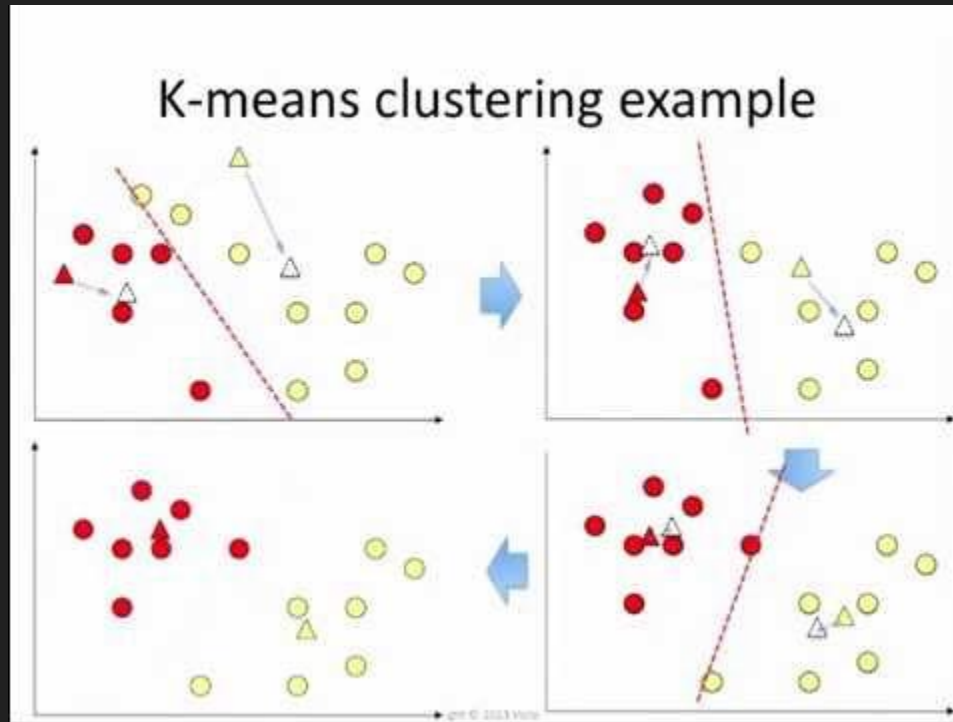
$h_2(x)$

...

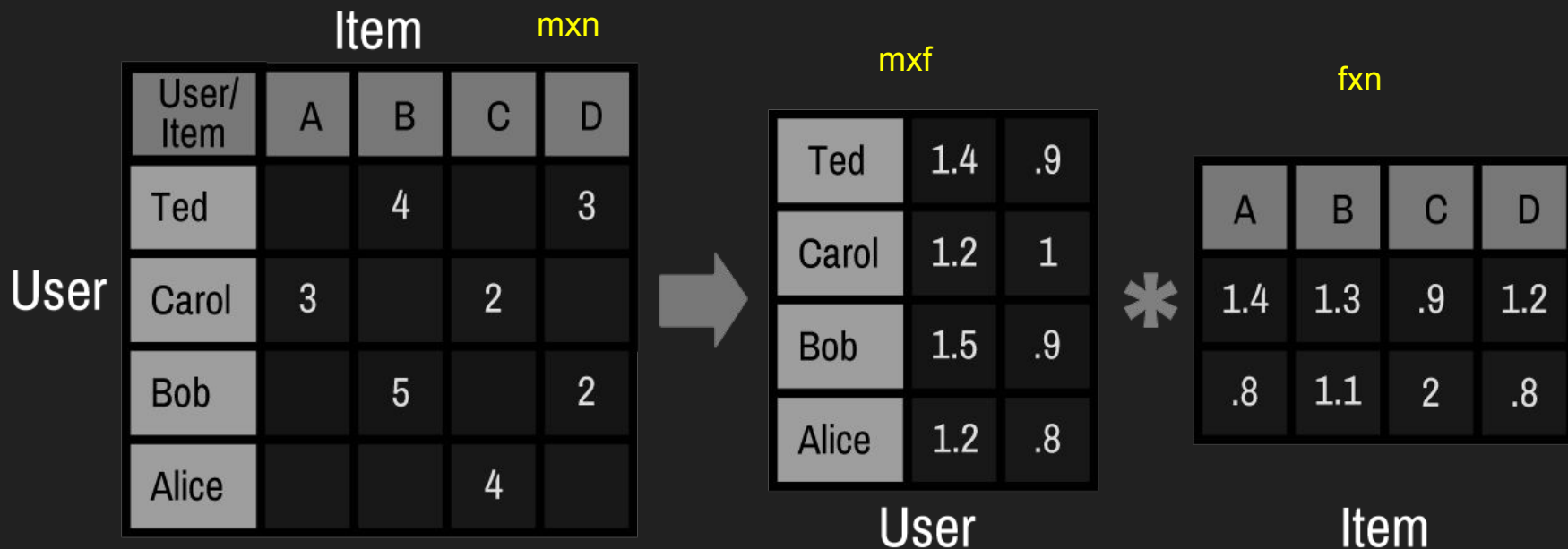


$h_n(x)$

KMeans



Collaborative Filtering



TED $A = 1.4 * 1.4 + .8 * .9 = 2.68$
 $B = 1.4 * 1.3 + .9 * 1.1 = 2.81$
 $C = 3.06$
 $D = 2.4$

Trained using Alternating Least Squares (ALS):
<http://yifanhu.net/PUB/cf.pdf>

Part 3 Tools for ML

Some Open Source ML Library Contributors



- => Many free open source ml libraries with high quality
- => No need to write everything from scratch

Status of ML Computation

More data && larger models => better models

Distribute



Accelerators







=> Difficult and time consuming to write everything from scratch





What prog. language is most used for data science?

Your primary programming language for Analytics, Data Mining, Data Science tasks: [512 voters]

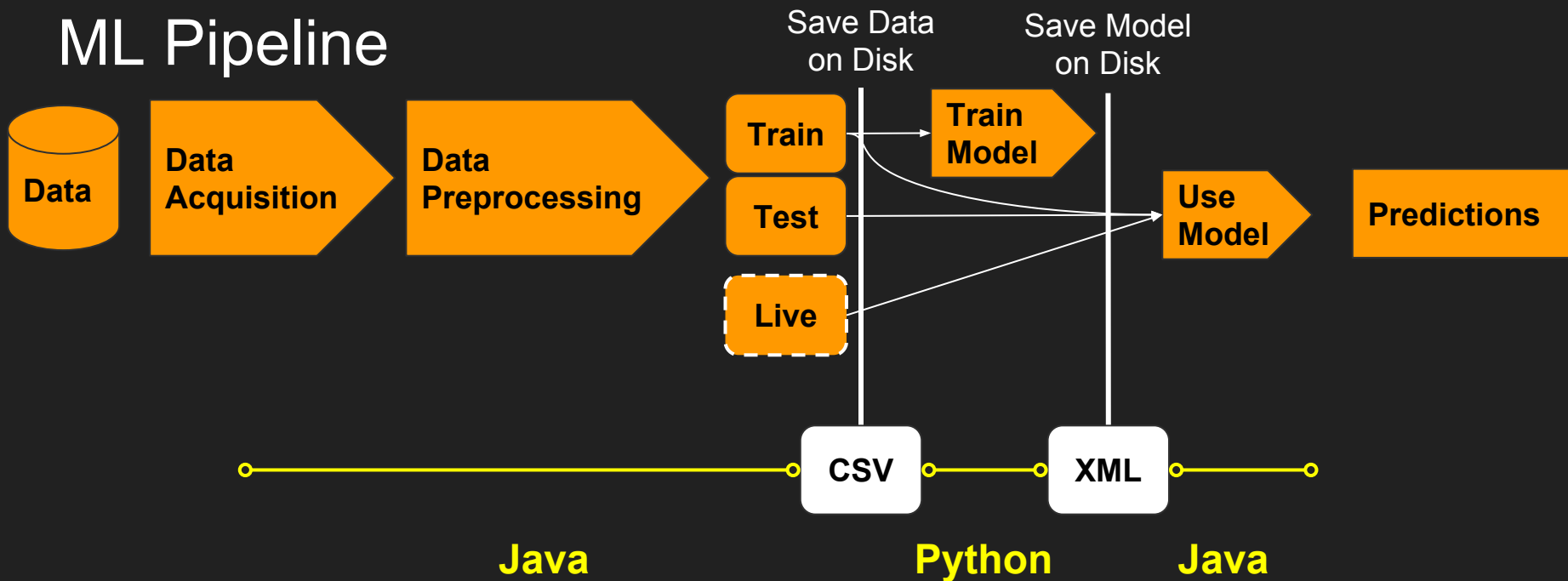
2015 primary programming language:

R (and its packages) (263)	 51% (of 2015 votes)
Python (including scikit-learn and other libraries) (151)	 29%
Other (Java, MATLAB, SAS, Scala, etc) (89)	 17%
none (9)	 1.8%

2014 primary programming language:

R (and its packages) (237)	 46% (of 2014 votes)
Python (including scikit-learn and other libraries) (117)	 23%
Other (Java, MATLAB, SAS, Scala, etc) (118)	 23%
none (40)	 7.8%

ML Pipeline



mongoDB
java-driver

MyJavaCleaner



“Python is slow”

Python

```
a = 0.1
x = FloatVector(100)
y = FloatVector(100)

x_sum = x.sum()
y.add(a, x)
```

Libraries

C/C++
Fortran



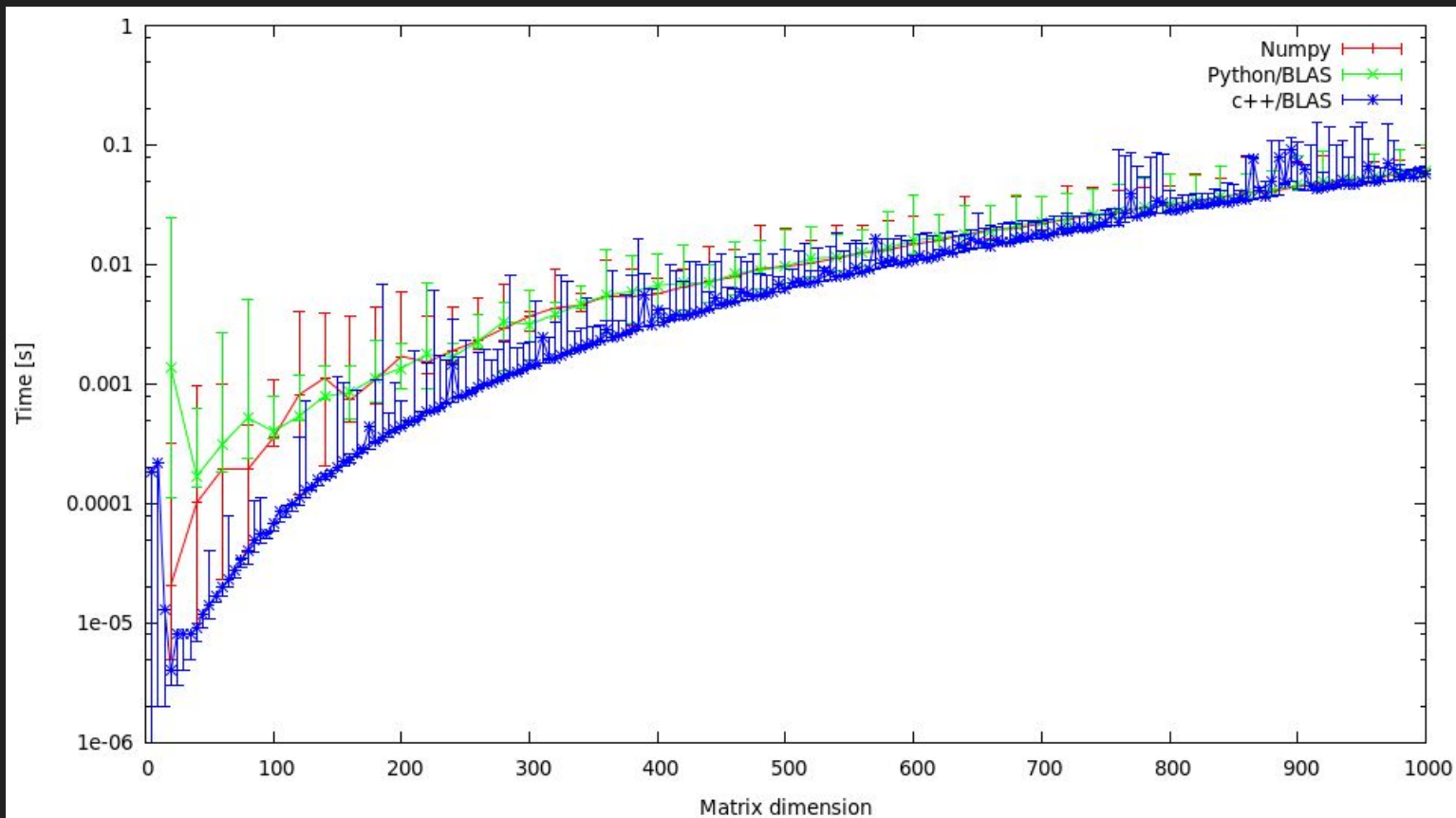
CUDA



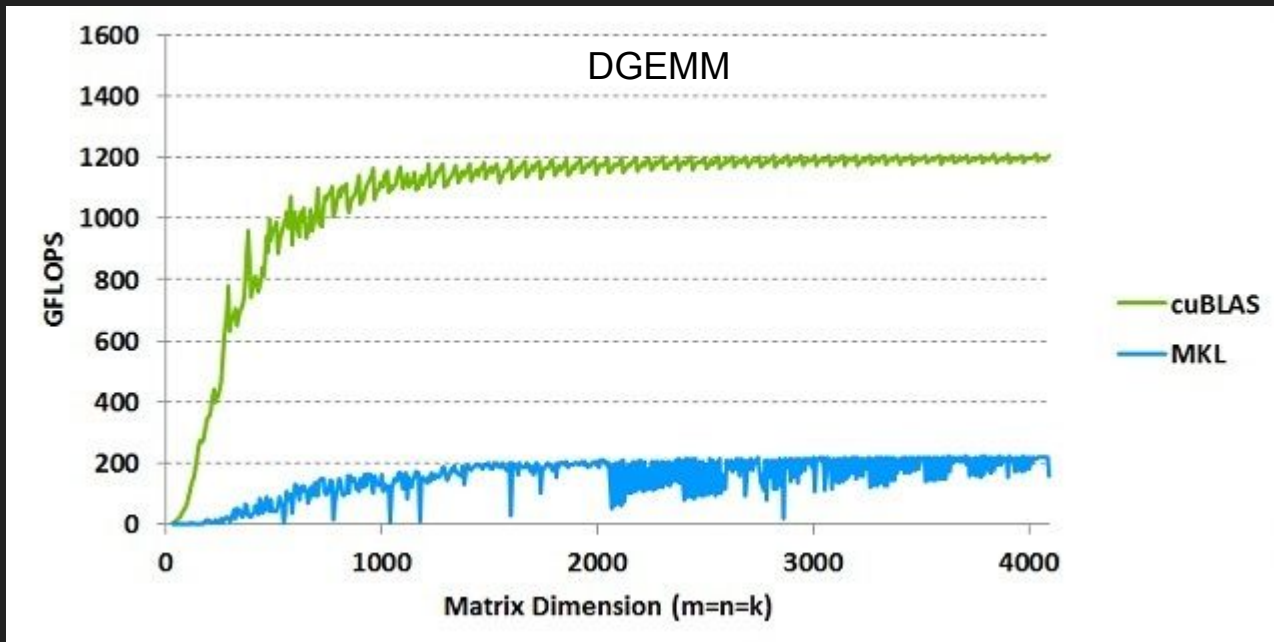
Your Code

```
class FloatVector {
    float* array;
    FloatVector(int size)
        array = new float[size];
    float sum()
        for (...) {}
    void add(float a, Vector& x)
        saxpy(...);
}
```

“Python is slow!”

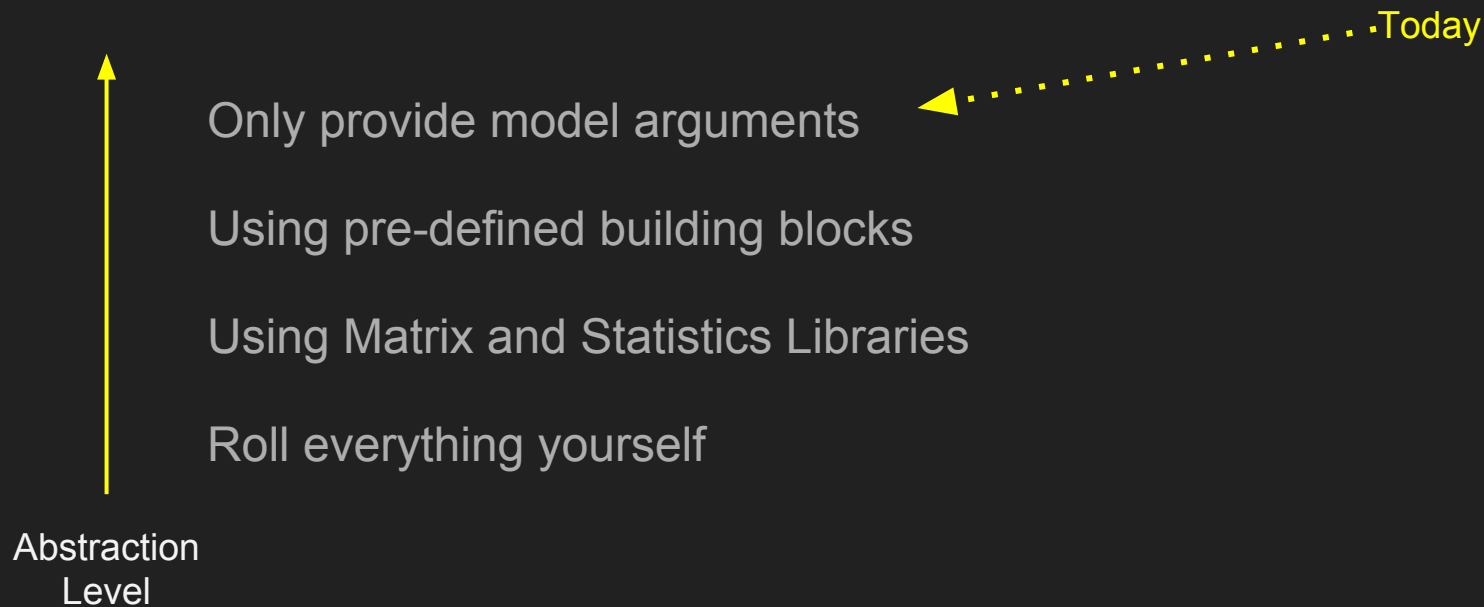


“Python is slow!”



cuBLAS on K40m, ECC ON, input and output data on device. MKL 11.0.4 on Intel IvyBridge single socket 12 -core E5-2697 v2 @ 2.70GHz

Abstraction Level



How easy it can be with Python and scikit-learn

mm_rain	temp	humidity	month
1.2	17	78	5
3.1	16	84	5
...

```
mm_rain,temp,humidity,month  
1.2,17,78,5  
3.1,16,84,5  
.....
```

1. Load csv/json/xml/txt/bin
2. Split into train and test set
3. Create Model
4. Train model
5. Evaluate model
6. Persist model

```
data = np.genfromtxt(path_to_csv,  
                      delimiter=',', names=True)  
X, y = data[:, 1:], data[:, 0]  
X_tr, X_te, y_tr, y_te = train_test_split(X, y, 0.2)  
model = SVC(C=1.0, kernel='rbf')  
model.fit(X_tr, y_tr)  
accuracy = model.score(X_te, y_te)  
joblib.dump(model, 'filename.pkl')
```

Machine learning in Spark

`spark.mllib` contains the original API built on top of RDDs.

`spark.ml` provides higher-level API built on top of DataFrames for constructing ML pipelines.

Data types

- Supports local vectors and matrices stored on a single machine, as well as distributed matrices backed by one or more RDDs.
- Numpy arrays and python lists are recognized as local dense vectors and MLlib's SparseVector and SciPy's csc_matrix as local sparse vectors.
- Local Matrix

```
import org.apache.spark.mllib.linalg.{Matrix, Matrices}

// Create a dense matrix ((1.0, 2.0), (3.0, 4.0), (5.0, 6.0))
dm2 = Matrices.dense(3, 2, [1, 2, 3, 4, 5, 6])

// Create a sparse matrix ((9.0, 0.0), (0.0, 8.0), (0.0, 6.0))
sm = Matrices.sparse(3, 2, [0, 1, 3], [0, 2, 1], [9, 6, 8])
```

Data types cont.

Distributed Matrices

- A **RowMatrix** is a row-oriented distributed matrix without meaningful row indices, backed by an RDD of its rows, where each row is a local vector.
- An **IndexedRowMatrix** is similar to a RowMatrix but with meaningful row indices. It is backed by an RDD of indexed rows, so that each row is represented by its index and a local vector.
- A **CoordinateMatrix** is a distributed matrix backed by an RDD of its entries. Each entry is a tuple of (i: Long, j: Long, value: Double), where i is the row index, j is the column index, and value is the entry value.
- A **BlockMatrix** is a distributed matrix backed by an RDD of MatrixBlocks, where a MatrixBlock is a tuple of ((Int, Int), Matrix), where the (Int, Int) is the index of the block, and Matrix is the sub-matrix at the given index with size rowsPerBlock x colsPerBlock.

LabeledPoint for Supervised Learning

```
from pyspark.mllib.linalg import SparseVector
from pyspark.mllib.regression import LabeledPoint

# Create a labeled point with a positive label and a dense feature vector.
pos = LabeledPoint(1.0, [1.0, 0.0, 3.0])
                    y      x

# Create a labeled point with a negative label and a sparse feature vector.
neg = LabeledPoint(0.0, SparseVector(3, [0, 2], [1.0, 3.0]))
                    y      x
```

Mllib in a slide

- logistic regression and linear support vector machine (SVM)
- classification and regression tree
- random forest and gradient-boosted trees
- recommendation via alternating least squares (ALS)
- clustering via k-means, bisecting k-means, Gaussian mixtures (GMM), and power iteration clustering
- topic modeling via latent Dirichlet allocation (LDA)
- survival analysis via accelerated failure time model
- singular value decomposition (SVD) and QR decomposition
- principal component analysis (PCA)
- linear regression with L_1 , L_2 , and elastic-net regularization
- isotonic regression
- multinomial/binomial naive Bayes
- frequent itemset mining via FP-growth and association rules
- sequential pattern mining via PrefixSpan
- summary statistics and hypothesis testing
- feature transformations
- model evaluation and hyper-parameter tuning

Exercise 1

1. Find a new dataset (regression or classification)

::USAGE:: see [file download_data.ipnb](#)
from sklearn.datasets import load_iris
data = load_iris()

<code>load_boston ()</code>	Load and return the boston house-prices dataset (regression).
<code>load_iris ()</code>	Load and return the iris dataset (classification).
<code>load_diabetes ()</code>	Load and return the diabetes dataset (regression).
<code>load_digits ([n_class])</code>	Load and return the digits dataset (classification).
<code>load_linnerud ()</code>	Load and return the linnerud dataset (multivariate regression).

Or find dataset from <http://mldata.org/>

2. Train a gradient boosted trees model with default parameters. Verify result on a testset. Similar to linear and logistic reg.. Guide at [http://spark.apache.org/docs/latest/mllib-ensembles.html#Gradient-Boosted-Trees-\(GBTS\)](http://spark.apache.org/docs/latest/mllib-ensembles.html#Gradient-Boosted-Trees-(GBTS))
3. Try some different combinations of hyper-parameter settings (learningRate and maxDepth) and choose the parameter settings with the best score on a validation set (a held out part of the training set). Verify result on same test set as above.

Exercise 2

1. Train a movie recommender system. Use dataset (ratings.csv) from yesterday and append some ratings of movies you have seen. You can follow this guide: <https://databricks-training.s3.amazonaws.com/movie-recommendation-with-mllib.html>
2. From the model, extract the user features (model.userFeatures()) and cluster them in 20 clusters. Where did you end up? Can you find something interesting from that cluster (you might need to join with other files, e.g. Movies.csv to get title and genre etc.).

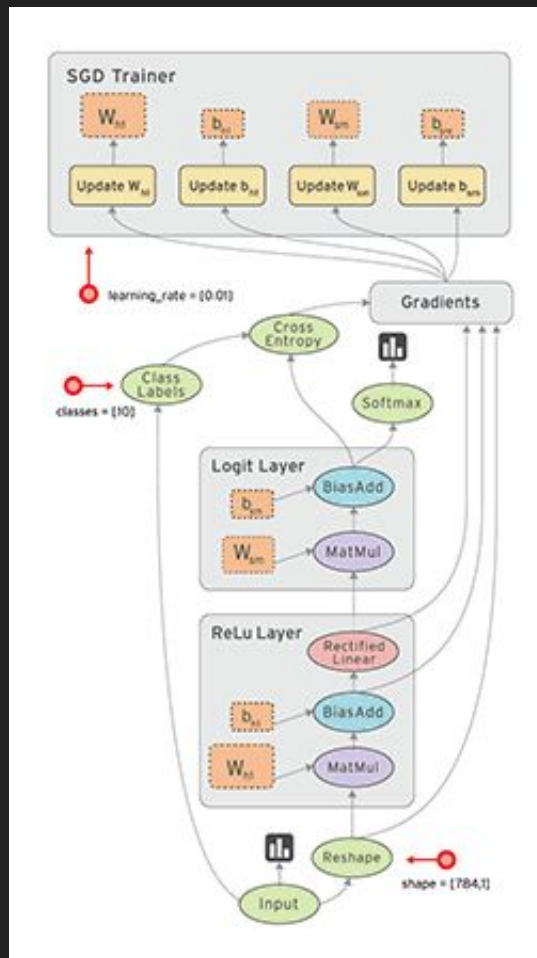
Linear Regression Demo

Homework for Friday

Think about how the tools that we have talked about this week can be used for your research. Prepare 2 min presentation till Friday (informal, no slides).

TensorFlow

- **Nodes** are operations
- **Directed edges** show the flow of data



Start with these nodes:

- `tf.placeholder(...)`
- `tf.Variable(...)`
- `tf.constant(...)`

Connect them via operations:

- `MatMul`
- `Add`
- `Reshape`
- `Slice`
- ...

Define **loss function**
and **Optimizer**

Execute Graph!