

Hands-on materials

Valentin Kuznetsov, Cornell University

SOSC 2019

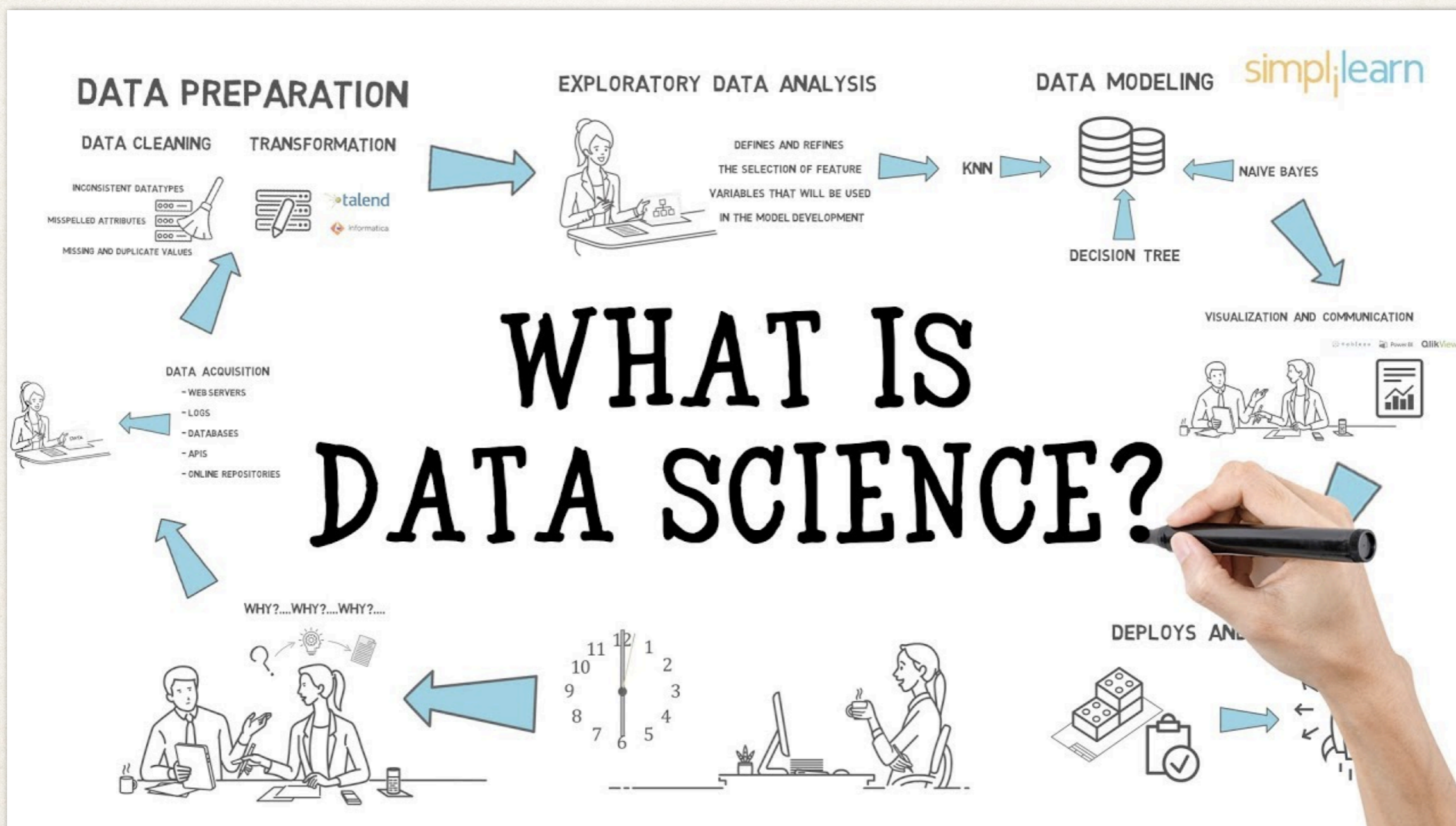
Who am I?

- ❖ Theoretical Physicists (neutrino oscillations) at Irkutsk Univ & JINR
- ❖ Particle Physicists (tracking, silicon detectors) at CERN
- ❖ PhD in Physics (theory + experiment) at JINR
- ❖ Computing in HEP at JINR, CERN, Fermilab, Cornell
 - ❖ HEP experiments: NOMAD, D0, Cleo-c, CMS
- ❖ Data Scientists at Cornell University
 - ❖ Data management, data discovery, services
 - ❖ BigData, Analytics, Monitoring, Machine Learning

Topics

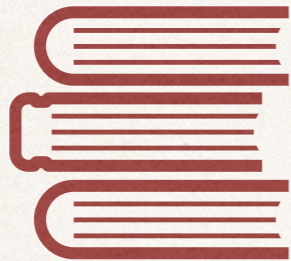
- ❖ Introduction
 - ❖ DataScience, DataScientists and Kaggle
- ❖ Day 1: setting up Data Science environment
- ❖ Day 2: dive into ML models
- ❖ Day 3: how to become a DataScientist (coverage of kaggle competition)
- ❖ Day 4: Image classification, training on GPUs/TPUs

Data Science



Data Scientist

Math,
Statistics,
Algorithms



iterate

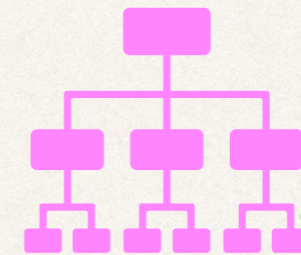


Computers
Programming

Tools
Compiles
Linkers
ML

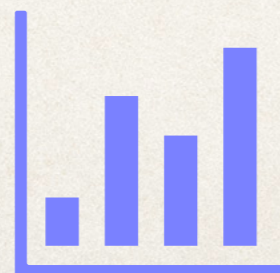


**Data
Scientists**



Clusters
Clouds
Platforms

Visualization
Data Analysis



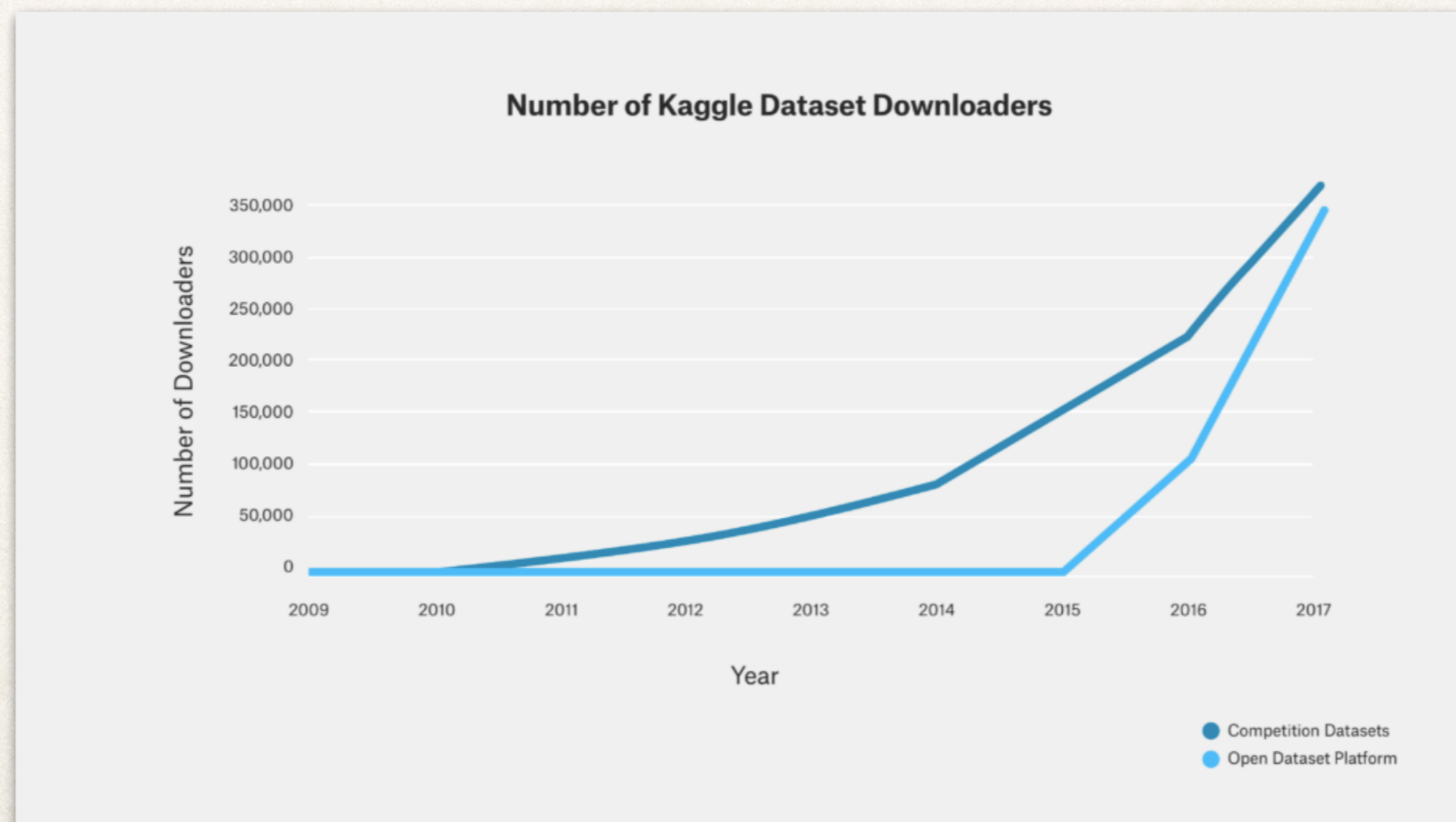
be global



Databases
SQL
No-SQL

kaggle.com

- ❖ It is open platform for Data Scientist to compete over published datasets
- ❖ In 2017: 120K DataScientists compete in 44 competitions, a total prize sum was \$4.75M+, 600K new users joined, 1.3M total users



My kaggle profile





- ❖ My goal is to learn ML / DL / AI and not prizes
 - ❖ my main source of DataScience
- ❖ I competed alone in my free time apart from regular job, teachings, student projects, family, etc.
 - ❖ turns out it is much tougher to compete alone since amount of info, data, training, ideas significantly increases
 - ❖ I was mostly active around 2015

The screenshot shows a Kaggle profile for a user named 'vk.net'. The profile includes a profile picture, location (Ithaca, NY, USA), and a 'Competitions Expert' badge. The user's current rank is 2226 out of 91,104, and their highest rank is 185. They have 1 gold medal, 1 silver medal, and 4 bronze medals. The profile also shows a list of competition results, including 'Acquire Valued Shoppers C...' (8th of 952), 'Otto Group Product Classifi...' (147th of 3514), and 'Tradeshift Text Classification' (68th of 375). The user is also listed as a 'Kernels Novice' and a 'Discussion Novice', both currently unranked. The user has 1 follower and an 'Edit Profile' button.
















Category	Rank	Medals	Details
Competitions Expert	Current Rank: 2226 of 91,104; Highest Rank: 185	1 Gold, 1 Silver, 4 Bronze	Acquire Valued Shoppers C... (8th of 952), Otto Group Product Classifi... (147th of 3514), Tradeshift Text Classification (68th of 375)
Kernels Novice	Unranked	0 Gold, 0 Silver, 0 Bronze	No kernel results
Discussion Novice	Unranked	0 Gold, 0 Silver, 11 Bronze	to kaggle team (4 votes), to kaggle team (3 votes), Public Start Guide of Deep ... (3 votes)

Kaggle Competitions

The screenshot shows the Kaggle website's 'Competitions' page. At the top, there is a navigation bar with the Kaggle logo, a search bar, and links for Competitions, Datasets, Kernels, Discussion, and Learn. Below the navigation bar, the 'Competitions' section is highlighted in blue, with buttons for 'Documentation' and 'InClass'. The main content area has tabs for 'General' and 'InClass', and a 'Sort by' dropdown menu set to 'Prize'. A search bar for competitions is also present. Below this, a blue banner indicates '298 Competitions'. The main list displays four featured competitions, each with a logo, title, description, featured status, date, tags, prize amount, and number of teams.

Logo	Competition Title	Description	Featured	Date	Tags	Prize	Teams
	Passenger Screening Algorithm Challenge	Improve the accuracy of the Department of Homeland Security's threat recognition algorit...	Featured	9 months ago	terrorism, image data, object detection	\$1,500,000	518 teams
	Zillow Prize: Zillow's Home Value Prediction (Zestimate)	Can you improve the algorithm that changed the world of real estate?	Featured	8 months ago	housing, real estate	\$1,200,000	3,779 teams
	Data Science Bowl 2017	Can you improve lung cancer detection?	Featured	a year ago	healthcare, image data, binary classification	\$1,000,000	1,972 teams
	Heritage Health Prize	Identify patients who will be admitted to a hospital within the next year using historical cl...	Featured	5 years ago		\$500,000	1,353 teams

Kaggle Competitions I did

	Acquire Valued Shoppers Challenge Predict which shoppers will become repeat buyers <i>Featured</i> · 4 years ago	  8/952 Top 1%
	Otto Group Product Classification Challenge Classify products into the correct category <i>Featured</i> · 3 years ago · 📁 internet, tabular data	  147/3514 Top 5%
	Tradeshift Text Classification Classify text blocks in documents <i>Featured</i> · 4 years ago	  68/375 Top 19%
	National Data Science Bowl Predict ocean health, one plankton at a time <i>Featured</i> · 3 years ago · 📁 oceanography, image data, multiclass classification	  90/1049 Top 9%
	Homesite Quote Conversion Which customers will purchase a quoted insurance plan? <i>Featured</i> · 2 years ago · 📁 tabular data, binary classification	  102/1764 Top 6%

❖ [Homesite](#) competition

1,764 Teams
1,939 Competitors
36,387 Entries

Homesite dataset

- ❖ Using an anonymized database of information on customer and sales activity, including property and coverage information, Homesite is challenging you to predict which customers will purchase a given quote. Accurately predicting conversion would help Homesite better understand the impact of proposed pricing changes and maintain an ideal portfolio of customer segments.
- ❖ This dataset represents the activity of a large number of customers who are interested in buying policies from Homesite. Each QuoteNumber corresponds to a potential customer and the QuoteConversion_Flag indicates whether the customer purchased a policy.
- ❖ The provided features are anonymized and provide a rich representation of the prospective customer and policy. They include specific coverage information, sales information, personal information, property information, and geographic information. Your task is to predict QuoteConversion_Flag for each QuoteNumber in the test set.
- ❖ Train sample: 299 columns (28 categorical variables), 260K rows (200MB); test sample 174K rows (131MB)

Homesite leaderboard

1	—	KazAnova Faron clobber		0.97024
2	—	Frenchies		0.97018
3	▲1	New Model Army CAD & QuY		0.97001
4	▼1	Gilberto Leustagos Stanislav		0.96988
5	—	The Northern Hemisphere		0.96983
6	▲1	victor, clustifier & adam		0.96968
7	▼1	monkeys rising		0.96961
8	—	A Few with NO Clue		0.96960
9	▲2	Daniel FG		0.96959
10	▼1	VinaKago		0.96956
100	▼11	BMX		0.96793
101	▲23	Overfitters		0.96792
102	▲101	vk.net		0.96792
		All Zeros Benchmark		0.50000

1st place

-0.49%

SOSC Mini-Kaggle

[Home](#) [Dashboards:](#) [Public](#) [Private](#)

MINI-KAGGLE SUBMISSION PAGE

Submission name

File name

 No file chosen

Submit

© [Valentin Kuznetsov](#) 2019

- ❖ During the school you will try to build your best model and submit it to our local mini-kaggle server
- ❖ We'll use Homesite (subset) dataset which is split into public (70%) and private (30%) ones
- ❖ Your submission will be evaluated on both using AUC score but only public score will be visible to you
- ❖ At the end of the school we'll release private scores and name a winner

```
ssh -L 8888:kaggle:8888 soscuser01@193.204.89.102
```


Before we start

Nodes

Sosc 2019 ☆ 🗑️

File Edit View Insert Format Data Tools Add-ons Help

🖨️ 🔍 100% 📄 View only

	A	B	C	D	E
1	VM NAME	PUBLIC IP	PRIVATE IP	Enabled username	
2	sosc19-01	193.204.89.102	172.16.0.63	soscuser01	
3	sosc19-01p		172.16.0.65	soscuser01	
4	sosc19-02	193.204.89.103	172.16.0.66	soscuser02	
5	sosc19-02p		172.16.0.67	soscuser02	
6	sosc19-03	193.204.89.108	172.16.0.69	soscuser03	
7	sosc19-03p		172.16.0.68	soscuser03	
8	sosc19-04	193.204.89.70	172.16.0.76	soscuser04	
9	sosc19-04p		172.16.0.77	soscuser04	
10	sosc19-05	193.204.89.119	172.16.0.70	soscuser05	
11	sosc19-05p		172.16.0.78	soscuser05	
12	sosc19-06	193.204.89.120	172.16.0.71	soscuser06	
13	sosc19-06p		172.16.0.79	soscuser06	
14	sosc19-07	193.204.89.121	172.16.0.72	soscuser07	
15	sosc19-07p		172.16.0.80	soscuser07	
16	sosc19-08	193.204.89.123	172.16.0.74	soscuser08	
17	sosc19-08p		172.16.0.81	soscuser08	
18	sosc19-09	193.204.89.122	172.16.0.73	soscuser09	
19	sosc19-09p		172.16.0.82	soscuser09	
20	sosc19-10	193.204.89.124	172.16.0.75	soscuser10	
21	sosc19-10p		172.16.0.83	soscuser10	
22	sosc19-11	193.204.89.79	172.16.0.84	soscuser11	
23	sosc19-11p		172.16.0.103	soscuser11	
24	sosc19-12	193.204.89.93	172.16.0.93	soscuser12	
25	sosc19-12p		172.16.0.94	soscuser12	
26	sosc19-13	193.204.89.80	172.16.0.85	soscuser13	

[Link](#)

School datasets

[Ref](#)

- ❖ [Iris dataset](#) will be used in ML introduction Hands-on
- ❖ [Homesite dataset](#) from kaggle will be used in our mini-kaggle competition
 - ❖ you must sign-up to kaggle and agree with competition rules
- ❖ [MNIST dataset](#) will be used for image classifications
- ❖ HEP image dataset may be used as a challenge
- ❖ You may look-up and download datasets [here](#)

Request TPU quota

Dear TFRC team,
could you please provide me an access to TPU resources on your platform as a part of SOSC 2019 school [1].

The TPU HandsOn session of the school will be carry on by Valentin Kuznetsov from Cornell University.

Name: <PUT YOUR NAME HERE>

Institute: <PUT YOUR INSTITUTE HERE, e.g. INFN Bologna>

Research domin: <PUT YOUR RESEARCH DOMAIN HERE, e.g. High Energy Physics>

Research interests: <PLEASE PROVIDE SHORT DESCRIPTION OF YOUR RESEARCH AREA>

Best regards,
<YOUR NAME HERE>

[1] <https://web.infn.it/SOSC19>

Hands-on session I

Setting up Data Science Environment
[Materials](#)

Hands-on session II

Working with basic ML models

[Materials](#)

Hands-on session III

How to become a Data Scientists

[Materials](#)

Recipe

- ❖ Node/environment setup
 - ❖ introduction to Anaconda
- ❖ Data exploration
 - ❖ introduction to R
- ❖ System limitations
 - ❖ issues with python, R and others
- ❖ Data preprocessing
 - ❖ intro to Python tools, common format, data scaling, normalization, working with NAs, etc.
- ❖ Training and modeling
- ❖ Reaching the limit
 - ❖ one-hot-encoding, leave-one-out encoding, word embeddings, ensembles, stacking, etc.

Embeddings recipe

- ❖ Identify categorical variables and order them
- ❖ Define embedded matrix and cardinality of categorical variable
- ❖ Perform one-hot-encoding
- ❖ Train Neural Network model
- ❖ Extract NN weights (embeddings matrix)
- ❖ Plug embeddings matrix into regular ML model instead of categorical variable
- ❖ Train ML model with embeddings matrices

One-hot-encoding

[Ref](#)

- ❖ It is a technique to handle “categorical” data
- ❖ It represents categorical column as vector of words
- ❖ You need to define word vector for full set of data (train + test datasets)
- ❖ Issues with NULL or missing data
 - ❖ delete rows with missing data
 - ❖ impute data for missing values

“One-Hot” refers to a state in electrical engineering where all of the bits in a circuit are 0, except a single bit with a value of 1 (it is said to be “hot”).

Rome	=	[1, 0, 0, 0, 0, 0, ..., 0]
Paris	=	[0, 1, 0, 0, 0, 0, ..., 0]
Italy	=	[0, 0, 1, 0, 0, 0, ..., 0]
France	=	[0, 0, 0, 1, 0, 0, ..., 0]

Leave-one-out encoding

[Ref](#)

- ❖ Use mean of all values within the same category except given row
- ❖ Add random noise
- ❖ Replace categorical value with leave-one-out times noise
- ❖ The test categorical values always represented as mean and no noise
- ❖ This technique may complement one-hot encoding

Split	UserID	Y	mean_y	random	new_Y
Train	A1	0			
Train	A1	1			
Train	A1	1			
Train	A1	0			
Test	A1	-			
Test	A1	-			
Train	A2	0			

We'll show how to add new categorical encoding features:

mean_y: average value for given user

random: random factor for given user

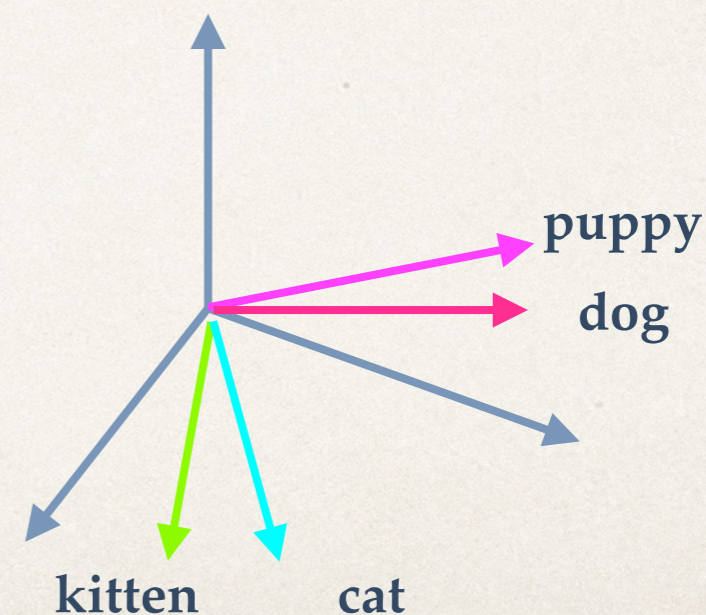
new_Y: new encoding feature for given user

Word embedding

[Ref](#)

- ❖ A way to capture multi-dimensional relationships between categories
 - ❖ e.g. cats/dogs can be called as puppies/kitten which represent their internal age
 - ❖ you define a dimension of word vector up-front
 - ❖ it projects categorical variables into another phase space, e.g. days may be sunny or rainy, season or off season; all of these features are hidden from original data representation
- ❖ Use NN or other ML algorithms to train the model to find best representation of embedded variables

real \ hidden	Dog Age Cat
puppy	[0.9, 1.0, 0.0]
dog	[1.0, 0.2, 0.0]
kitten	[0.0, 0.1, 0.9]
cat	[0.0, 1.0, 1.0]



Stacking

[Ref](#)

- ❖ Split original dataset into 3 datasets: A, B, C
- ❖ For A and B we know the ground truth
- ❖ We train various ML algorithms on dataset A
- ❖ We make predictions of our ML models on datasets B and C and we create new datasets B1 and C1 that only contains predictions, so if we run 10 algorithms we'll have 10 columns in B1 and C1 datasets
- ❖ We train new Meta ML algorithm using dataset B1
- ❖ We make predictions using Meta ML model on dataset C1

Consider datasets A,B,C. Target variable (y) is known for A,B...

Hands-on session IV

Image Classification on GPUs
[Materials](#)

Training on GPUs/TPUs

- ❖ Using fast.ai, see [fastai_example.py](#)
 - ❖ requires minimal learning curve; produces top-world results out of the box
- ❖ Using PyTorch, see [pytorch_examples.py](#)
 - ❖ flexible Python-like interface building your NN/DL networks; dynamic data-model
- ❖ Using TensorFlow, see [keras_dn.py](#)
 - ❖ industry standard and plenty of usage in production; static-graphs model; scalability and inference deployment
- ❖ Using TF on TPUs, see [keras_tpu_101.py](#), [keras_dn_tpu.py](#)
 - ❖ next level of scalability on dedicated hardware

Welcome to fastai

The fastai library simplifies training fast and accurate neural nets using modern best practices. It's based on research in to deep learning best practices undertaken at [fast.ai](#), including "out of the box" support for [vision](#), [text](#), [tabular](#), and [collab](#) (collaborative filtering) models. If you're looking for the source code, head over to the [fastai repo](#) on GitHub. For brief examples, see the [examples](#) folder; detailed examples are provided in the full documentation (see the sidebar). For example, here's how to train an MNIST model using [resnet18](#) (from the [vision example](#)):

```
path = untar_data(URLs.MNIST_SAMPLE)
data = ImageDataBunch.from_folder(path)
learn = cnn_learner(data, models.resnet18, metrics=accuracy)
learn.fit(1)
```

Total time: 00:09

epoch	train_loss	valid_loss	accuracy
1	0.128580	0.082647	0.973503

PyTorch vs TensorFlow

[Ref](#)

- ❖ **dynamic vs static graph definitions**

- ❖ RNNs implementation: with static graphs the input sequence length will stay constant, i.e. TF has limited support for dynamic inputs

- ❖ **debugging:** in PyTorch you can stop and inspect your model using python debugger, e.g. pdb, while for TF you need a special tool which will evaluate expressions

- ❖ **visualization:** TF has awesome TensorBoard tool which can be used to visualize network, inspect hyper-parameters, etc.

- ❖ **deployment:** TF supports multiple languages, can be deployed as gRPC server, support mobile, etc. While for PyTorch we may use python web framework, e.g. Flask, and develop specialized REST APIs

- ❖ **data parallelism:** PyTorch smoothly parallelize over data batches (declarative data parallelism), while TF allows to tune/run every operation on dedicated device. Both allows to run on multiple GPUs, but TF can run on TPUs

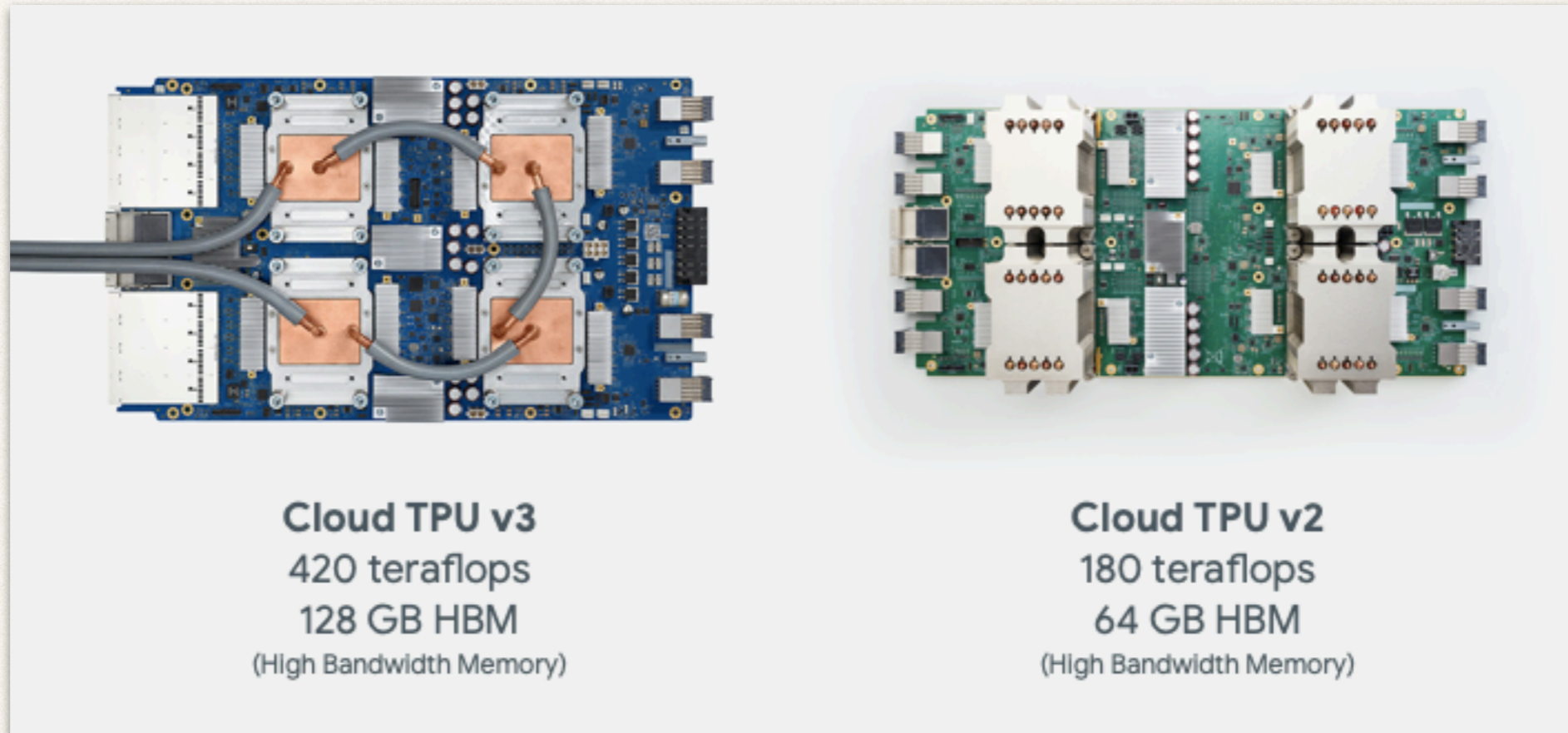
- ❖ framework (PyTorch) vs data-library (TF)

Hands-on session V

Image Classification on TPUs
[Materials](#)

TPUs

[Ref](#)



TPUs are dedicated hardware developed by Google which is fine-tuned for various ML and Deep Learning tasks. It is designed for matrix multiplications along with vector processor. Its specs are represented in terms of Matrix Multiple Unit (MXU) and Vector Processing Unit (VPU). The former operates in 16-32 bit floats while latter handles float32 and int32 computations.

In Google data centers TPUs are grouped in TPU pods which consist of 512 TPU v2 cores connected through HPC interconnect.