Fundamentals and Landscape of Classical Machine Learning (II)

- Learn the framing of supervised learning
- Know the modern machine learning landscape
- Learn the basics of Scikit-Learn

https://www.sli.do/
#073374

Learn the framing of supervised learning

Supervised Machine Learning

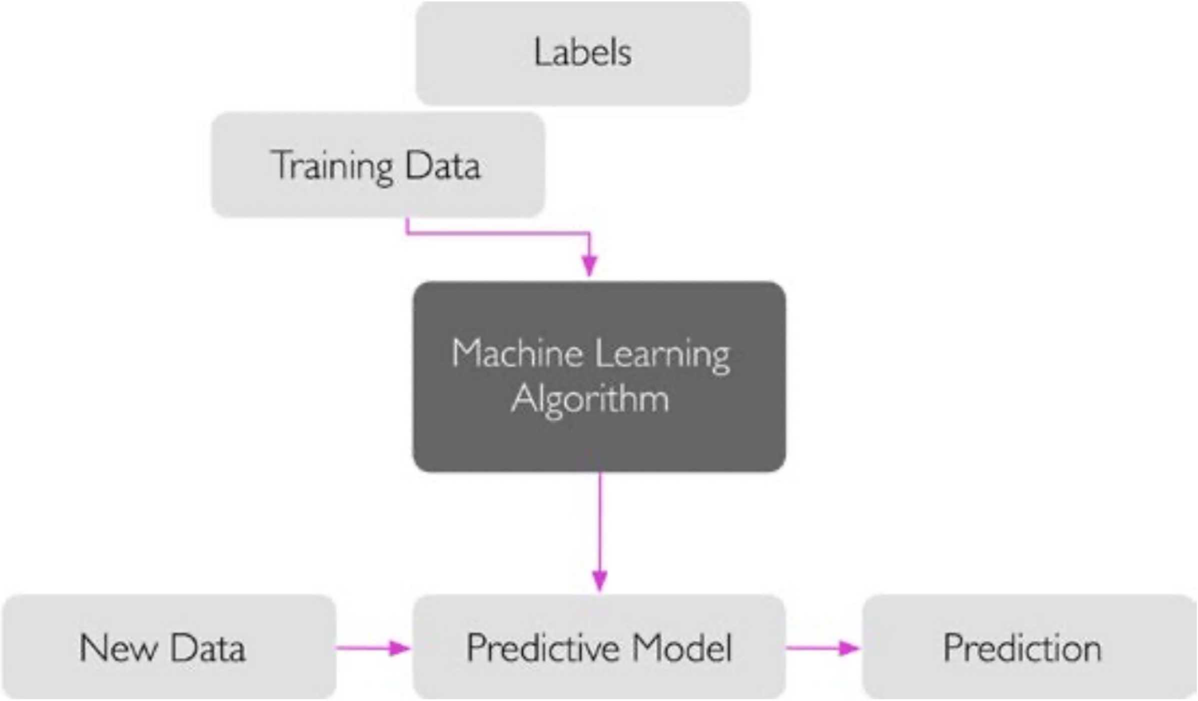
credit: Google, machine learning crash course,
https://developers.google.com/machine-learning/crash-course

ML models learn

how to combine input

to produce useful predictions

on never-before-seen data



Supervised Machine Learning

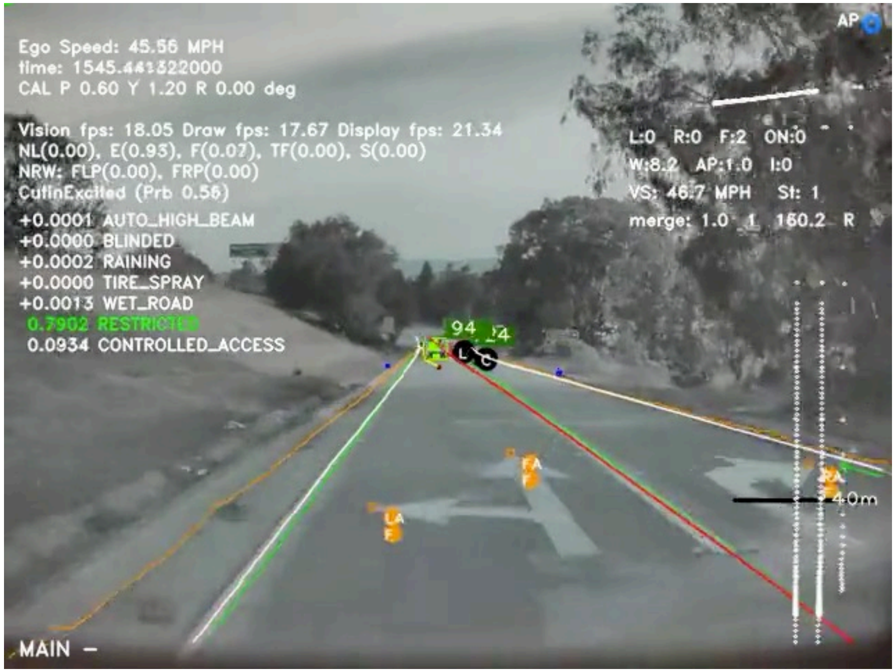
credit: Google, machine learning crash course,
https://developers.google.com/machine-learning/crash-course

ML models learn

how to combine input

to produce useful predictions

on never-before-seen data



```
Ego Speed: 45.56 MPH
time: 1545.441322000
CAL P 0.60 Y 1.20 R 0.00 deg

Vision fps: 18.05 Draw fps: 17.67 Display fps: 21.34
NL(0.00), E(0.93), F(0.07), TF(0.00), S(0.00)
NRW: FLP(0.00), FRP(0.00)
CutinExcited (Prb 0.56)
+0.0001 AUTO_HIGH_BEAM
+0.0000 BLINDED
+0.0002 RAINING
+0.0000 TIRE_SPRAY
+0.0013 WET_ROAD
 0.7902 RESTRICTED
 0.0934 CONTROLLED_ACCESS
```

```
L:0   R:0   F:2   ON:0
W:8.2   AP:1.0   I:0
VS: 46.7 MPH     St: 1
merge: 1.0  1  160.2  R
```

https://www.tesla.com/autopilotAI

(The deep neural) networks learn from the most complicated and diverse scenarios in the world, iteratively sourced from our fleet of nearly 1M vehicles in real time. A full build of Autopilot neural networks involves 48 networks that take 70,000 GPU hours to train 🔥. Together, they **output 1,000 distinct tensors (predictions)** at each timestep.

Supervised Machine Learning Terminology

- **Label** is the variable we're predicting
  - Typically represented by the variable y
- **Features** are input variables describing our data
  - Typically represented by the variables $\{x_1, x_2, \dots, x_D\}$
- **Example** is a particular instance of data, $\boldsymbol{x}$ (**bold** indicates a vector)
- **Labeled example** has {features, label}: $\{\boldsymbol{x}, y\}$
  - Used to train the model
- **Unlabeled example** has {features, ?}
  - Used to making prediction on new data
- **Model** maps unlabeled examples to predicted labels: y'
  - Defined by (training) parameters, which are learned.

credit: Google, machine learning crash course,
https://developers.google.com/machine-learning/crash-course

Supervised Machine Learning (Credit Approval)

Labeled examples

age	gender	annual	year in	year in	current	approval
		salary	residence	job	debt	
(feature)	(feature)	(feature)	(feature)	(feature)	(feature)	(label)
23	female	1,000,000	1	0.5	200,000	Yes
45	male	500,000	1	0.5	250,000	No
75	male	0	20	0	0	Yes

Unlabeled examples

age	gender	annual	year in	year in	current
		salary	residence	job	debt
(feature)	(feature)	(feature)	(feature)	(feature)	(feature)
45	female	1,500,000	10	5	500,000

Fun Time: Supervised Machine Learning

https://www.sli.do/
#073374

Suppose you want to develop a supervised machine learning model to predict whether a given email is “spam” or “not spam.” Which of the following statements are true? (多選題)

(i) Emails not marked as "spam" or "not spam" are unlabeled examples.

(ii) The labels applied to some examples might be unreliable.

(iii) We'll use unlabeled examples to train the model.

(iv) Words in the subject header will make good labels.

credit: Google, machine learning crash course,
https://developers.google.com/machine-learning/crash-course

Fun Time: Features and Labels

https://www.sli.do/
#073374

Suppose an online shoe store wants to create a supervised ML model that will provide personalized shoe recommendations to users. That is, the model will recommend certain pairs of shoes to Marty and different pairs of shoes to Janet. The system will use past user behavior data to generate training data. Which of the following statements are true? (多選題)

(i) "Shoes that a user adores" is a useful label.

(ii) "Shoe beauty" is a useful feature.

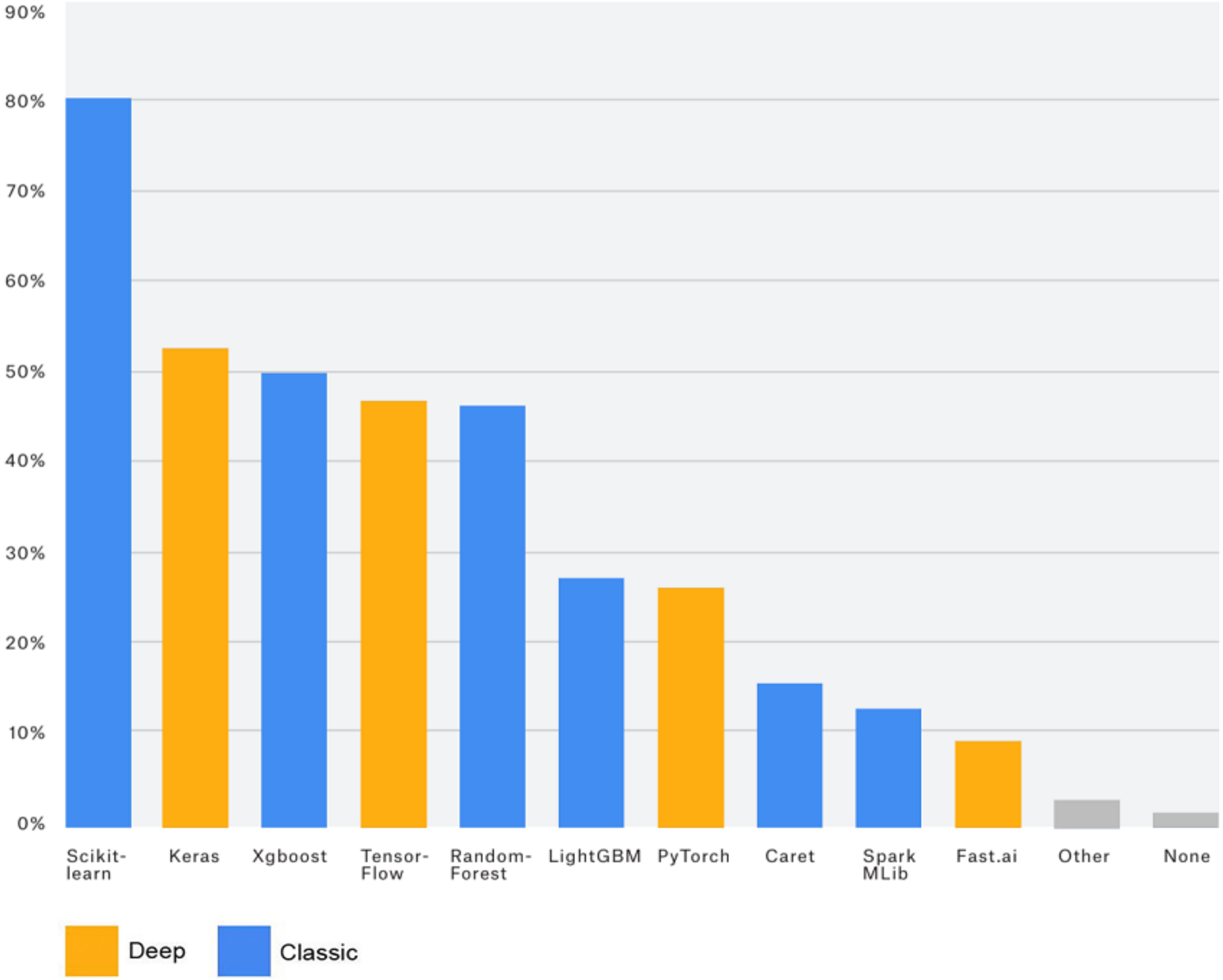
(iii) "The user clicked on the shoe's description" is a useful label.

(iv) "Shoe size" is a useful feature.

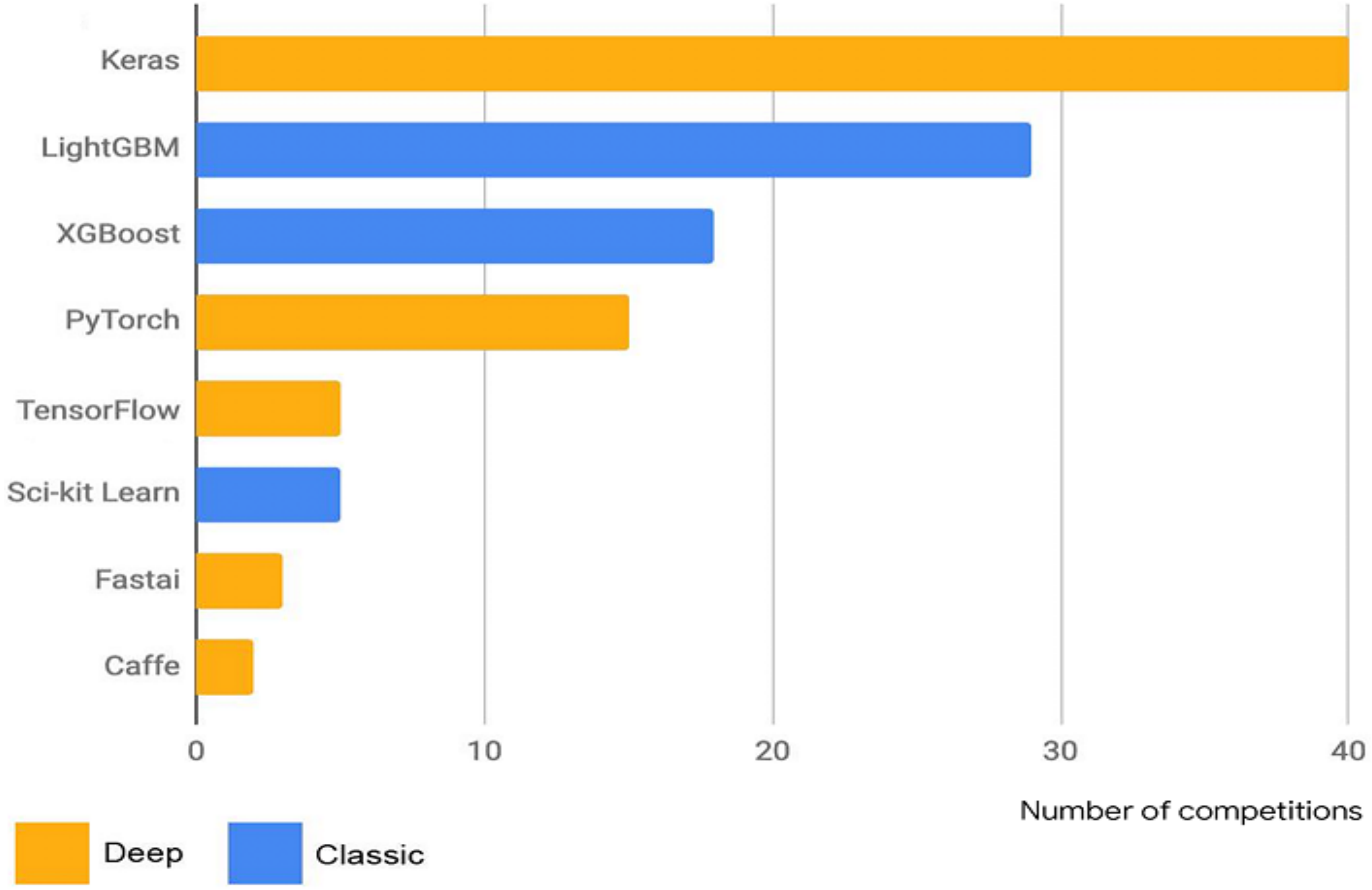
credit: Google, machine learning crash course,
https://developers.google.com/machine-learning/crash-course

Know the modern machine learning landscape

Percentage of machine learning & data science professionals using each ML software framework, 2019



Primary ML tool used by top-5 teams in Kaggle competitions, 2017-2018 (N=120)



Summary

Know the modern ML landscape

- Scikit-Learn and Keras (now part of TensorFlow) are mostly widely used ML software frameworks by ML professionals.

- From 2016 to 2020, the entire machine learning and data science industry has been dominated by these two approaches: deep learning and gradient boosted trees. Specifically, gradient boosted trees is used for problems where structured data is available, whereas deep learning is used for perceptual problems such as image classification.

- Users of gradient boosted trees tend to use Scikit-Learn, XGBoost or LightGBM. Meanwhile, most practitioners of deep learning use Keras, often in combination with its parent framework TensorFlow.

- The common point of these tools is they’re all Python libraries: Python has is by far the most widely-used language for machine learning and data science.

Learn the basics of Scikit-Learn

scikit learn

Machine Learning with Scikit-Learn

Extensions to **SciPy** (Scientific Python) are called **SciKits**. **SciKit-Learn** provides machine learning algorithms.

- **Algorithms for supervised & unsupervised learning**
- **Built on SciPy and Numpy**
- **Standard Python API interface**
- **Probably the best general ML framework out there.**

credit: Introduction to Machine Learning with Scikit-Learn, District Data Lab

scikit
learn

Machine Learning with Scikit-Learn

Primary Features

- Generalized Linear Models
- SVMs, kNN, Bayes, Decision Trees, Ensembles
- Clustering and Density algorithms
- Cross Validation
- Grid Search
- Pipelining
- Model Evaluations
- Dataset Transformation
- Dataset Loading

credit: Introduction to Machine Learning with Scikit-Learn, District Data Lab

scikit learn

Machine Learning with Scikit-Learn

Object-oriented interface centered around the concept of an Estimator:

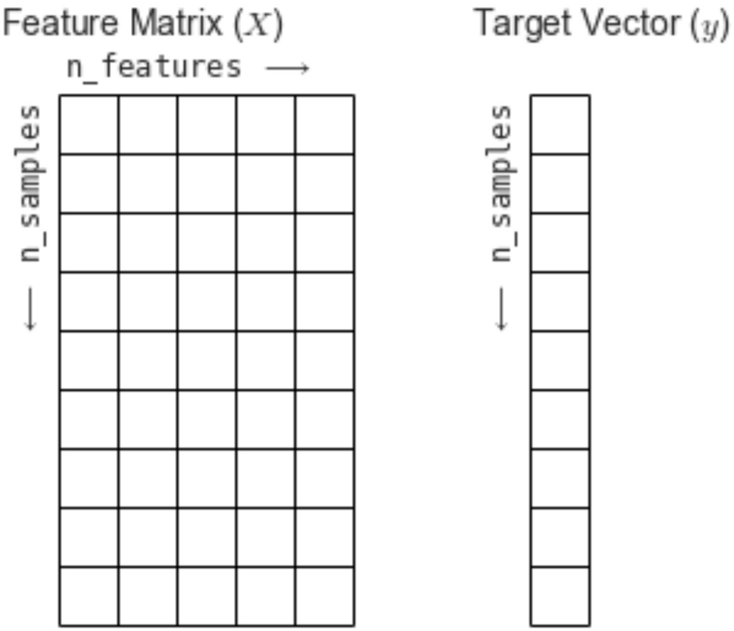
“**An estimator is any object that learns from data**; it may be a classification, regression or clustering algorithm or a transformer that extracts/filters useful features from raw data.”

credit: Introduction to Machine Learning with Scikit-Learn, District Data Lab

Machine Learning with Scikit-Learn

Estimators

- **fit(X,y)** sets the state of the estimator.
  - **X** is usually a 2D numpy array of shape (num_samples, num_features)
  - **y** is a 1D array with shape (n_samples,)
- **predict(X)** returns the class or value



See Introducing_Scikit-Learn.pdf

credit: Introduction to Machine Learning with Scikit-Learn, District Data Lab