

The background of the slide is a dark blue-tinted aerial photograph of a university campus, showing numerous buildings, green lawns, and trees.

outrageously
AMBITIOUS

Module 4: ML System Design & Technology Selection

Duke
PRATT SCHOOL OF
ENGINEERING

Module 4 Objectives:

At the conclusion of this module, you should be able to:

- 1) Describe the key technology decisions involved in designing ML systems
- 2) Identify the main criteria to consider in making technology selection decisions
- 3) Explain the commonly used tools among data scientists and ML practitioners

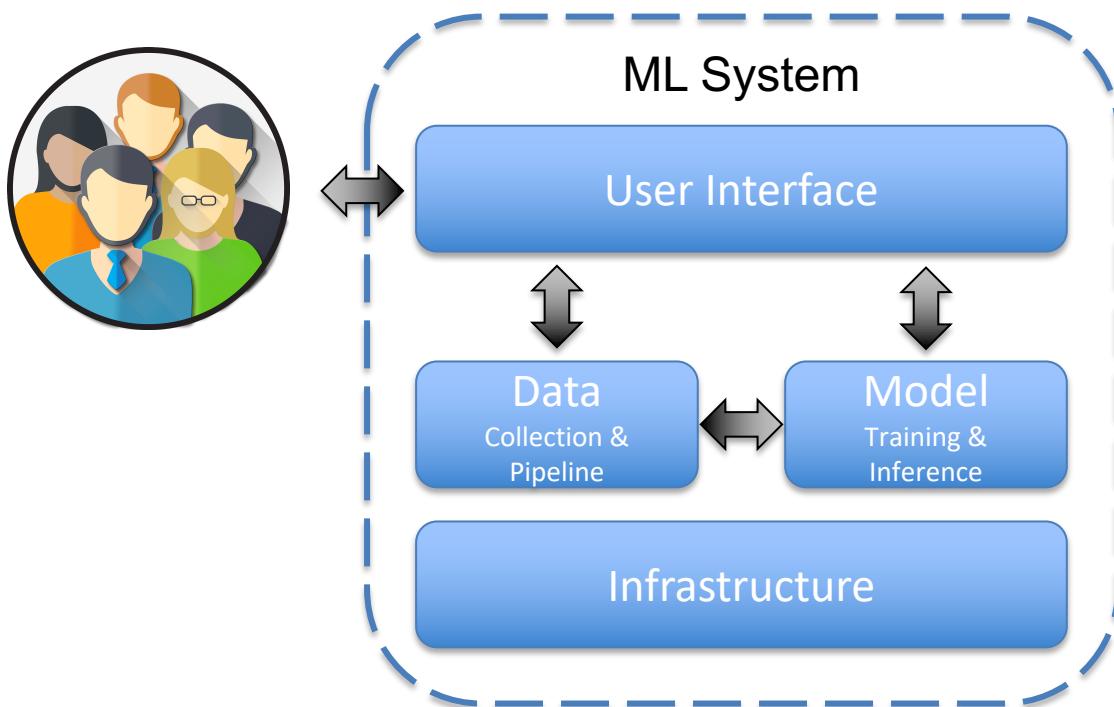
The background of the slide is a dark blue-tinted aerial photograph of a university campus. The campus features a mix of architectural styles, including several large, light-colored Gothic-style buildings with intricate stonework and multiple towers, and more modern, low-slung engineering and science buildings. The grounds are filled with green lawns and mature trees.

outrageously
AMBITIOUS

ML System Design Considerations

Duke
PRATT SCHOOL OF
ENGINEERING

What is a ML system?



ML system design decisions

There are several system design decisions which impact the choice of technologies:

Cloud

vs.

Edge

Offline
Learning

vs.

Online
Learning

Batch
Predictions

vs.

Online
Predictions

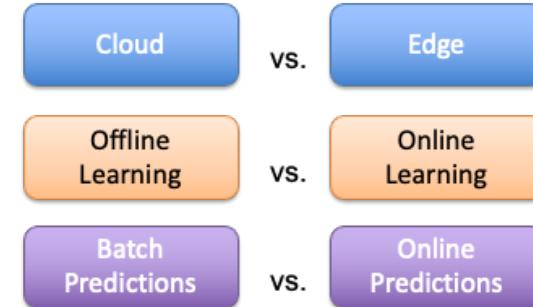
System design process

- User requirements and constraints drive system design
- System design drives selection of technologies

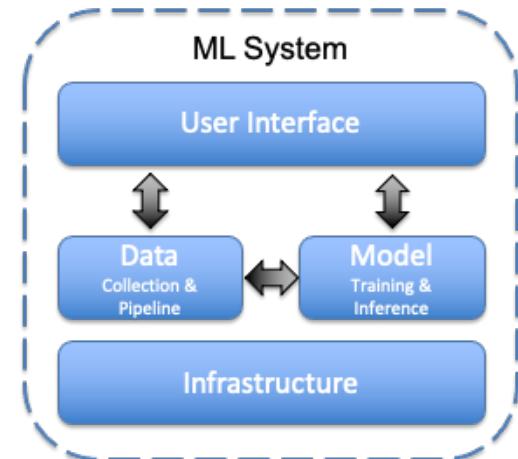
User requirements & constraints



System design decisions



Technology decisions



System design examples

Example 1

Use Case

- Unlocking a phone using a facial recognition model



Requirements / Constraints

- Low latency
- Cannot require connectivity
- Privacy concerns



System design

Cloud

vs.

Edge

Offline Learning

vs.

Online Learning

Batch Predictions

vs.

Online Predictions

Example 2

Use Case

- Movie recommendation engine



Requirements / Constraints

- Assume connectivity
- High throughput
- Low latency



Cloud

vs.

Edge

Offline Learning

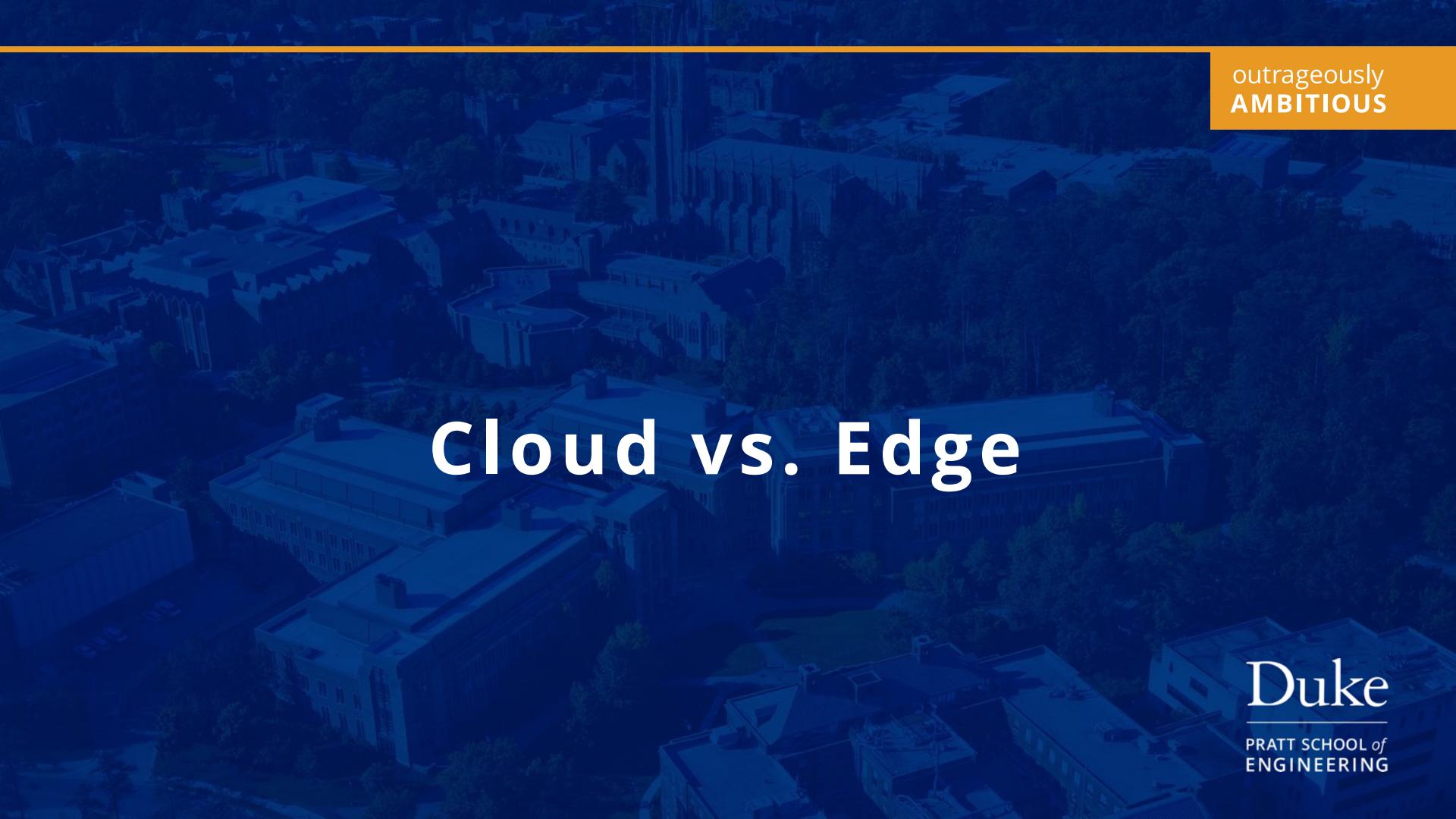
vs.

Online Learning

Batch Predictions

vs.

Online Predictions

The background of the slide is a high-angle aerial photograph of a university campus. The buildings are a mix of traditional Gothic-style structures with tall, thin spires and more modern, low-slung academic buildings. The campus is surrounded by a dense forest of green trees. A large, light-colored building with a prominent arched entrance is visible in the center-left.

outrageously
AMBITIOUS

Cloud vs. Edge

Duke

PRATT SCHOOL of
ENGINEERING

Edge AI

- Running ML on devices themselves
- Predicted to grow by 25% per year¹
- Enabled by hardware & software advances
 - Solving power constraints
 - Increased edge computational power
 - Smaller models designed for the edge
- Why edge AI? Eliminates latency
 - Every 100 miles distance from datacenter introduces a latency of > 1.6 milliseconds²

1) Valuates, <https://reports.valuates.com/market-reports/QYRE-Auto-4139/global-edge-ai-software>

2) Techwalla, <https://www.techwalla.com/articles/network-latency-milliseconds-per-mile>

Cloud vs. edge

	Cloud ML	Edge ML
Description	Computations done on cloud and result delivered to end device	Computations done directly on device (phone, sensor, etc)
Requirements	Network connectivity	Sufficient compute power, memory
Benefits	High throughput	Low latency, privacy, no need for connectivity
Examples	<ul style="list-style-type: none">• Chatbots• Demand prediction	<ul style="list-style-type: none">• Quality control• Autonomous driving

Cloud ML example

Movie recommendation system



Edge ML example

Intelligent security system



Hybrid approaches

- Initiate cloud ML through trigger generated by edge AI
- Store common pre-computed predictions on device
- Exploit many local datacenters/servers to minimize latency

Hybrid example

Smart speaker with voice assistant



Cloud vs. edge AI

- How much does latency matter?
 - A lot -> Edge
 - Not much -> Cloud
- Is reliance on internet connectivity acceptable?
 - No -> Edge
 - Yes -> Cloud
- Are users comfortable sending their data to the cloud?
 - No -> Edge
 - Yes -> Cloud

The background of the slide is a dark blue-tinted aerial photograph of a university campus. The campus features several large, historic Gothic-style buildings with intricate stonework and tall spires. Interspersed among these are more modern, low-slung engineering and science buildings with glass windows and industrial-looking roofs. Lush green trees and lawns cover much of the ground between the buildings.

outrageously
AMBITIOUS

Online Learning & Inference

Duke
PRATT SCHOOL of
ENGINEERING

Offline vs. online models

An important design consideration is whether model training & prediction can be scheduled or must be real-time

	Scheduled	Real-time
Model re-training	Offline learning	Online learning
Prediction	Batch prediction	Online prediction

Offline vs. online learning

	Offline learning
Description	Model re-training done on a schedule (weeks/months) using datapoints in many iterations
Benefits	<ul style="list-style-type: none">• Easier to implement in production• Easier to evaluate
Challenges	<ul style="list-style-type: none">• Slower to adapt to changes in environment or data distribution
Examples	<ul style="list-style-type: none">• Most current applications

Offline vs. online learning

	Offline learning	Online learning
Description	Model re-training done on a schedule (weeks/months) using datapoints in many iterations	Continual re-training as new data arrives (mins/hours) using each new datapoint once
Benefits	<ul style="list-style-type: none">Easier to implement in productionEasier to evaluate	<ul style="list-style-type: none">Handles big dataReal-time adaptation to changing environment
Challenges	<ul style="list-style-type: none">Slower to adapt to changes in environment or data distribution	<ul style="list-style-type: none">Harder to implement & evaluate performance
Examples	<ul style="list-style-type: none">Most current applications	<ul style="list-style-type: none">Flagging spam in social media

Online learning example

News site with personalized
recommendations



Batch vs. online prediction

	Batch prediction
Description	Generate predictions on batch of observations on a recurring schedule
Benefits	<ul style="list-style-type: none">• Leverage more efficient operations and technologies• Easier monitoring of drift
Challenges	<ul style="list-style-type: none">• Predictions not immediately available for new data
Examples	<ul style="list-style-type: none">• Recommendation systems• Demand prediction

Batch vs. online prediction

	Batch prediction	Online prediction
Description	Generate predictions on batch of observations on a recurring schedule	Real-time predictions generated upon request
Benefits	<ul style="list-style-type: none">Leverage more efficient operations and technologiesEasier monitoring of drift	<ul style="list-style-type: none">Predictions available immediately
Challenges	<ul style="list-style-type: none">Predictions not immediately available for new data	<ul style="list-style-type: none">Minimizing latencyMonitoring of model drift
Examples	<ul style="list-style-type: none">Recommendation systemsDemand prediction	<ul style="list-style-type: none">Translation appAutonomous vehicles

Online prediction example

Food delivery “time-to-arrive”



The background of the slide is a dark blue-tinted aerial photograph of a university campus. The campus features several large, historic Gothic-style buildings with intricate stonework and tall spires. Interspersed among these are more modern, low-slung engineering and science buildings with glass windows and metallic roofs. Lush green trees and lawns cover much of the ground between the buildings.

outrageously
AMBITIOUS

ML on Big Data

Duke
PRATT SCHOOL of
ENGINEERING

Examples of Big Data

Sensor/device data



Social media data



Healthcare data

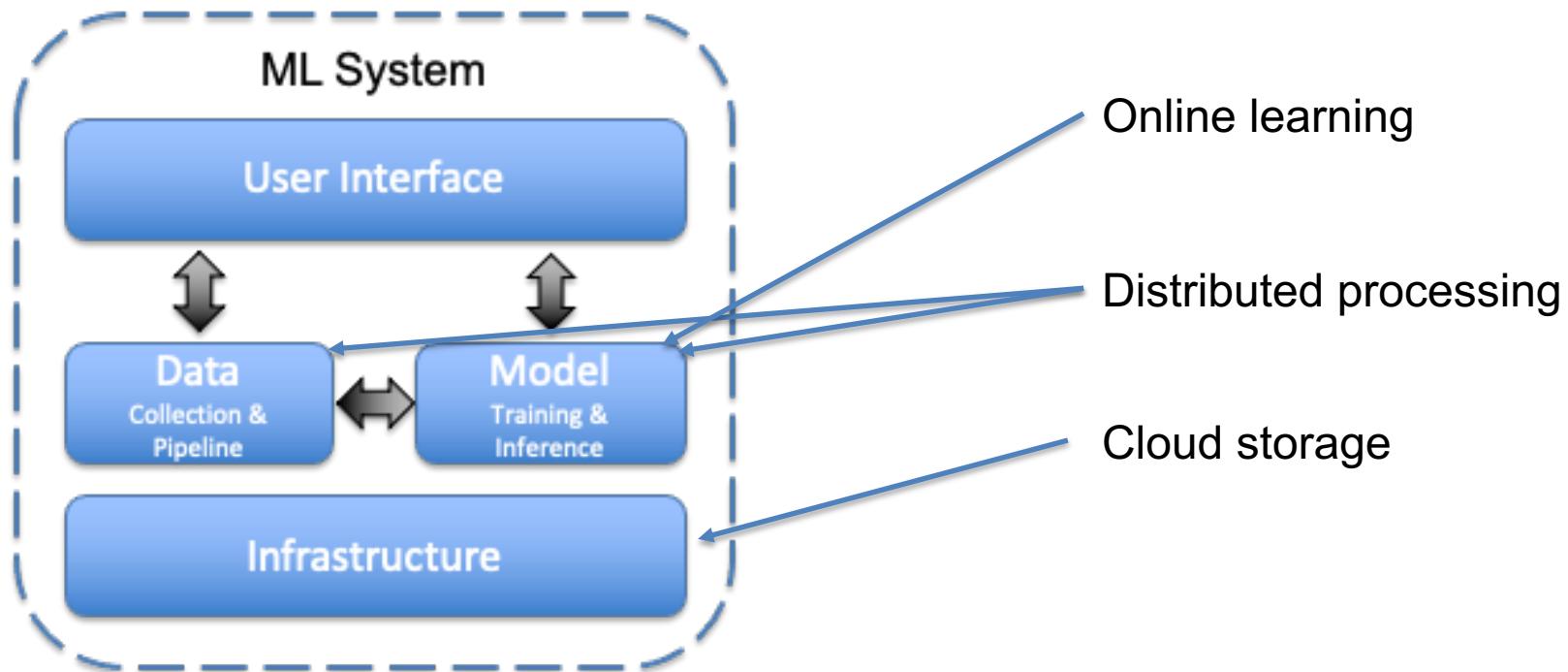


ML on Big Data

Working with Big Data has unique challenges:

- **Storage & processing**
 - Massive storage needs
 - Processing computational costs
- **Exploration**
 - Understanding quality, patterns & relationships
- **Modeling**
 - Data is too large to fit in memory
 - Long model training cycles

Impacts on ML system design



The background of the slide is a dark blue-tinted aerial photograph of a university campus. The campus features several large, historic Gothic-style buildings with intricate stonework and tall spires. Interspersed among these are more modern, low-slung engineering and science buildings with glass windows and metallic roofs. Lush green trees and lawns cover much of the ground between the buildings.

outrageously
AMBITIOUS

ML Technology Selection

Duke
PRATT SCHOOL of
ENGINEERING

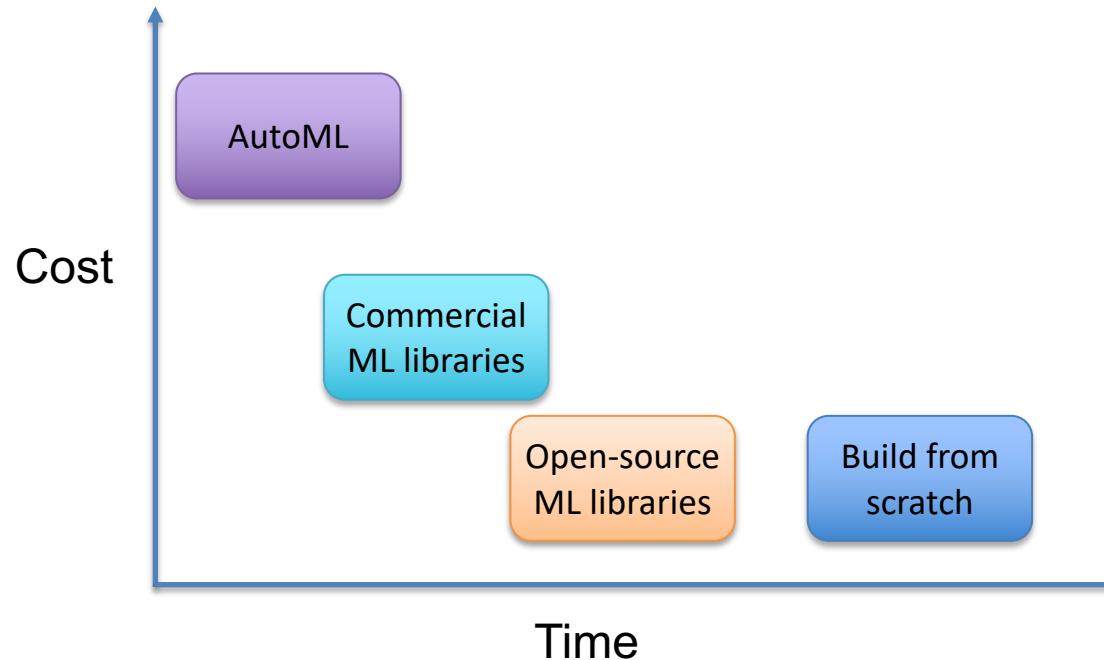
ML technology decisions

- Programming language
- Data processing toolset
- Modeling toolset
- API / interface

Factors in technology selection

- Open source vs. proprietary
- Learning curve
- Documentation – docs, tutorials etc.
- Community support & talent
- Flexibility vs. simplicity
- Lifetime cost

ML technology options



The background of the slide is a dark blue-tinted aerial photograph of a university campus. The campus features several large, historic buildings with red roofs and white walls, interspersed with modern glass and steel structures. Lush green trees and lawns cover much of the ground between the buildings.

outrageously
AMBITIOUS

Common ML Tools

Duke
PRATT SCHOOL of
ENGINEERING

Programming languages

Python

- Easy to learn, highly readable
- Rich ecosystem of data science / ML libraries
- Widely available help resources
- Very strong community

R

- Specialized for statistical and graphical analysis
- Good ecosystem of data analysis libraries
- Excellent plotting capabilities
- Good community

C/C++

- Common choice for software dev teams
- Faster than other languages
- Higher learning curve
- Less commonly used for ML

Language popularity

PYPL Popularity of Programming Index

Rank	Language	Share	1yr Trend
1	Python	30.6%	-1.1%
2	Java	17.2%	-0.1%
3	JavaScript	8.3%	+0.3%
4	C#	6.8%	0%
5	C/C++	6.3%	+0.6%
6	PHP	6.2%	+0.2%
7	R	3.9%	-0.1%

Source: <http://pypl.github.io/PYPL.html>

Jupyter notebooks

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** jupyter text_classification_examples_agnews Last Checkpoint: 4 hours ago (unsaved changes)
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, Python 3
- Cells:**
 - In [99]:** # Create features using TfIdf
tfidf=TfidfVectorizer()
X_train=tfidf.fit_transform(train_df['processed_text'])
X_test=tfidf.transform(test_df['processed_text'])
 - In [100]:** # Train a classifier using logistic regression classifier
y_train = train_df['Class_Index']
logreg_model = LogisticRegression(solver='saga')
logreg_model.fit(X_train,y_train)
preds = logreg_model.predict(X_train)
acc = sum(preds==y_train)/len(y_train)
print('Accuracy on the training set is {:.3f}'.format(acc))

Accuracy on the training set is 0.944
 - In [101]:** # Evaluate accuracy on the test set
y_test = test_df['Class_Index']
test_preds = logreg_model.predict(X_test)
test_acc = sum(test_preds==y_test)/len(y_test)
print('Accuracy on the test set is {:.3f}'.format(test_acc))

Accuracy on the test set is 0.917
- Section Headers:** Modeling using TFIDF features, Model using Word2Vec embeddings

- Great for experimentation
- Combine code (>40 languages), text and images in a single notebook
- Enable iterative development through individual cell execution
- More difficult to follow software engineering best practices – testing, versioning

Python ecosystem

- **Pandas (and NumPy)**: data manipulation
 - Merge, manipulate, clean & analyze data using DataFrames
- **Sci-kit Learn**: modeling
 - Data analysis and classical machine learning
- **Matplotlib**: visualization
 - Visualize data from pandas and NumPy
- **NLTK, SpaCy**: text processing
 - Pre-process text and convert to features for modeling

Deep learning libraries

TensorFlow

- Open-source Python libraries for building and training deep neural network models
- Can be used on CPUs, GPUs or TPUs for accelerated model training
- Export trained models to servers and devices for external use by applications

PyTorch

- Developed by Google
- More mature than PyTorch
- Historically the largest user community

Keras

- Developed by Facebook AI Research Lab
- Slightly easier learning curve than TensorFlow
- Growing popularity and user community

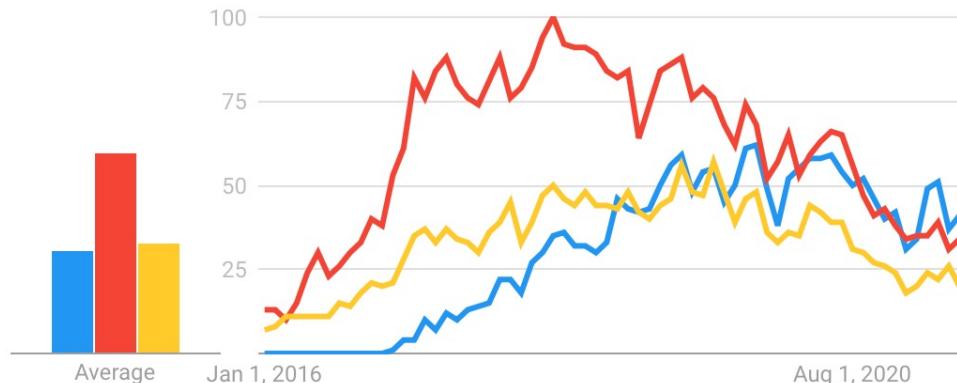
- Deep learning API on top of TensorFlow
- Simplifies model building process to enable quick iteration

Deep learning libraries

Interest over time

Google Trends

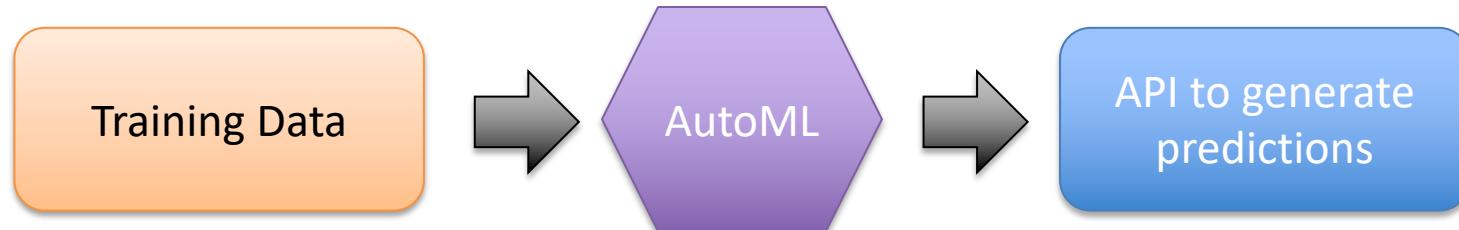
● pytorch ● tensorflow ● keras



Source: <https://trends.google.com/trends/explore?date=2016-01-01%202021-06-23&geo=US&q=pytorch,tensorflow,keras>

AutoML

- Enables developers with limited ML expertise to quickly build models with little/no code
- Automates selection of features, algorithm and hyperparameters for optimal performance
- Available from Google, Microsoft, AWS, H2O, etc.
- Variants for common ML tasks including NLP, computer vision



The background of the slide is a dark blue-tinted aerial photograph of the Duke University campus. The image shows a dense cluster of buildings, including several large Gothic-style structures, modern dormitories, and research facilities. The campus is surrounded by a mix of green trees and manicured lawns.

outrageously
AMBITIOUS

Wrap-up

Duke
PRATT SCHOOL of
ENGINEERING

Wrap Up

- User requirements -> ML system design -> technology selection
- Important considerations include latency, connectivity, throughput, data size
- Many technology options
 - Open source vs proprietary
 - Manual vs automated