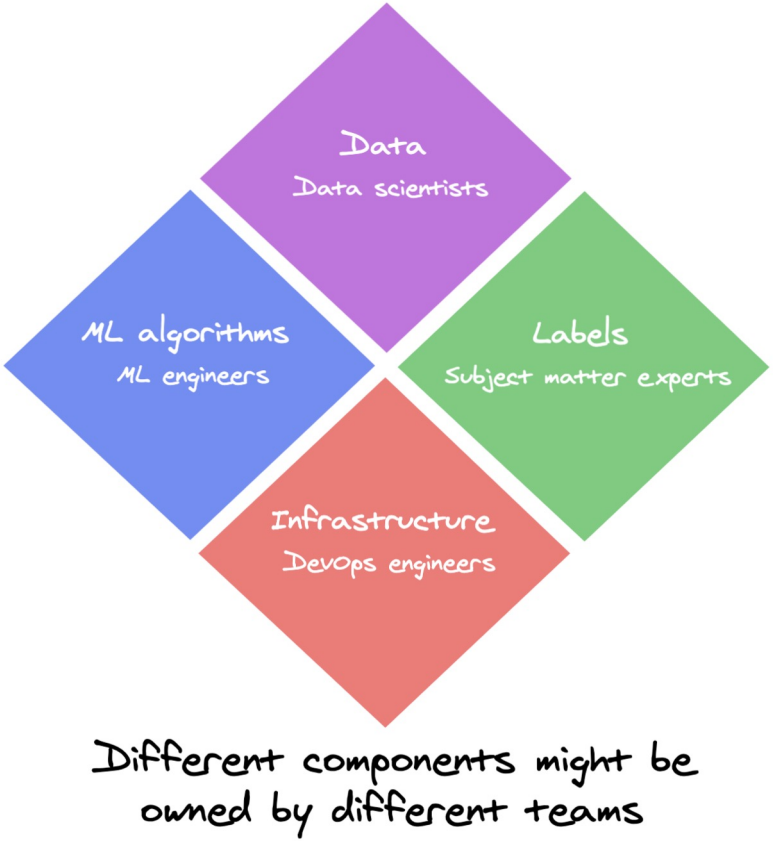
Evaluation and Testing of Machine Learning Systems

Why is debugging ML systems hard?

Cross-functional complexity

- Components owned by different teams
- Errors can be because of any or a combination



Why is debugging ML systems hard?

Silent errors

0 People Like You

Facebook translates 'good morning' into 'attack them', leading to arrest

Palestinian man questioned by Israeli police after embarrassing mistranslation of caption under photo of him leaning against bulldozer

Slide from: CS 329 | Chip Huyen | cs329s.stanford.edu

Why is debugging ML systems hard?

Slow updating cycle

Might have to wait a long time to see whether a bug is fixed

https://xkcd.com/303/

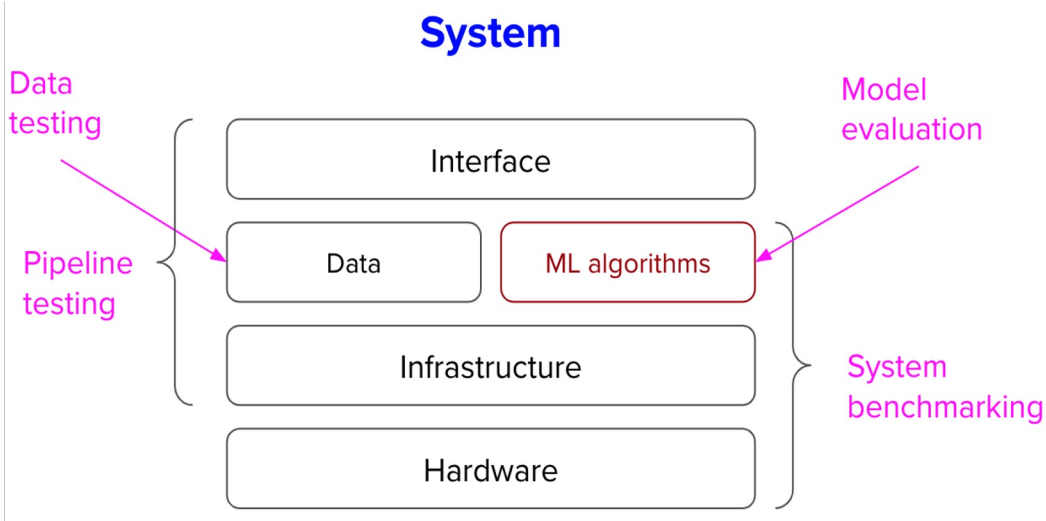
Sources of errors

- **Theoretical constraints**
  - Poor model/data fit, problems impossible to solve with ML, etc.
- **Poor model implementation/Poor choice of hyperparameters**
- **Data problems**
  - mismatched inputs/labels, over/under-preprocessed data, noisy labels, bad splits
- **Feature engineering problems**
  - Features lacking predictive power
  - Data leakage
- **Pipeline problems**
  - changes in training pipeline not replicated in inference pipeline

Slide from: CS 329 | Chip Huyen | cs329s.stanford.edu

ML system evaluation

- **Data testing**: ensuring new data satisfies your assumptions
  - **NOT covered here**: test to see how good a new sample is for your model
- **Pipeline testing**: ensuring your pipeline is set up correctly
- **~System benchmarking~**: reporting your system's overall performance on well-defined tasks, metrics, and rules, usually for the purpose of comparison
- **Model evaluation**: evaluating how good your ML model is



Slide from: CS 329 | Chip Huyen | cs329s.stanford.edu

Pipeline testing

- Consistency - Does your system make the same prediction on the same input
    - during training & inference?
    - on different runs?
- Integration between components - Can the entire pipeline run end-to-end from data loading to feature engineering to training to serving?
- [Hard] Visibility
    - Can you map a sample's transformation through your entire pipeline?
- Edge cases
    - What happens if inputs violate assumptions?

Slide from: CS 329 | Chip Huyen | cs329s.stanford.edu

Tips for ensuring your pipeline correctness

- Start simple and gradually add more components
  - Validate prediction assumption
  - Validate pipeline
  - Understand the impact of each added component
- Make sure an input has same output in both training and inference
- Write an end-to-end test
- Set random seeds
  - For reproducibility
- Change random seeds
  - The more sensitive your model is to inconsequential things, the harder it'll be retrain/maintain it

Slide from: CS 329 | Chip Huyen | cs329s.stanford.edu

Data testing: label quality & correlation

- Manually check quality of your labels
- If labels come from multiple sources, check correlation/conflicts among them

Slide from: CS 329 | Chip Huyen | cs329s.stanford.edu

Data testing: feature engineering

- Do features make sense?
- When new features are added, are old features still necessary?
- Are missing values replaced with mean/default values?
- Sanity-check after each transformation

Slide from: CS 329 | Chip Huyen | cs329s.stanford.edu

Data testing: assumptions

- Models embed assumptions about data
  - Both **inputs and outputs**
  - Example assumptions:
    - geotags are available -> model performs weird when geotags are null
    - answers' lengths are within 1 minute
- How to know if these assumptions hold in production?

Slide from: CS 329 | Chip Huyen | cs329s.stanford.edu

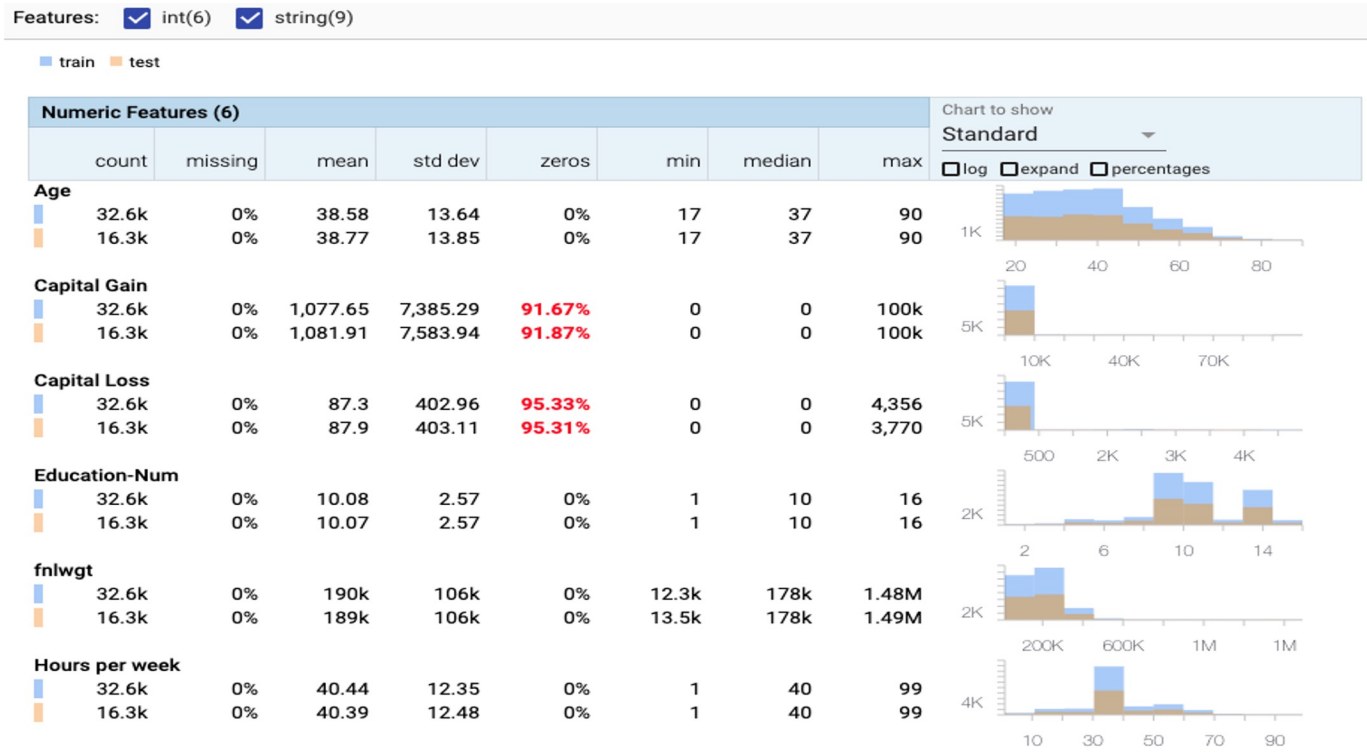
Data testing: assumptions

- Pre-train
  - Data exploration
    - Some data might make no sense to non-experts
  - **Data profiling & visualization**
  - **Create assertions**
- Post-train
  - Make sure each new data sample passes assertions

```
StartTime,Dur,Proto,SrcAddr,Sport,Dir,DstAddr,Dport,State,sTos,dTos,TotPkts,TotBytes,SrcBytes,Label
2011/08/18 10:21:46.633335,1.060248,tcp,93.45.239.29,1611,    ->,147.32.84.118,6881,S_RA,0,0,4,252,132,flow=Background-TCP-Attempt
2011/08/18 10:19:49.027650,279.349152,tcp,62.240.166.118,1031,  <?>,147.32.84.229,13363,SRPA_PA,0,0,15,1318,955,flow=Background-TCP-Attempt
2011/08/18 10:22:07.160628,166.390015,tcp,147.32.86.148,58067,    ->,66.235.132.232,80,SR_SA,0,0,3,212,134,flow=Background-TCP-Established
2011/08/18 10:26:02.052163,1.187083,tcp,147.32.3.51,3130,    ->,147.32.84.46,10010,S_RA,0,0,4,244,124,flow=Background-TCP-Attempt
2011/08/18 10:26:52.226748,0.980571,tcp,88.212.37.169,3134,    ->,147.32.84.118,6881,S_RA,0,0,4,244,124,flow=Background-TCP-Attempt
2011/08/18 10:27:57.611601,1.179357,tcp,94.44.60.103,49905,    ->,147.32.84.118,6881,S_RA,0,0,4,268,148,flow=Background-TCP-Attempt
2011/08/18 10:28:15.463038,1.140237,tcp,2.159.127.100,1378,    ->,147.32.84.118,6881,S_RA,0,0,4,252,132,flow=Background-TCP-Attempt
2011/08/18 10:28:37.132447,12.058948,tcp,213.233.154.219,36381,    ->,147.32.84.229,13363,SRA_SA,0,0,7,508,208,flow=Background-TCP-Established
2011/08/18 10:29:03.150806,0.942243,tcp,88.212.37.169,62055,    ->,147.32.84.118,6881,S_RA,0,0,4,244,124,flow=Background-TCP-Attempt
2011/08/18 10:29:43.750355,3.547213,tcp,95.210.161.212,58571,    ->,147.32.84.118,6881,S_RA,0,0,8,488,248,flow=Background-TCP-Attempt
```

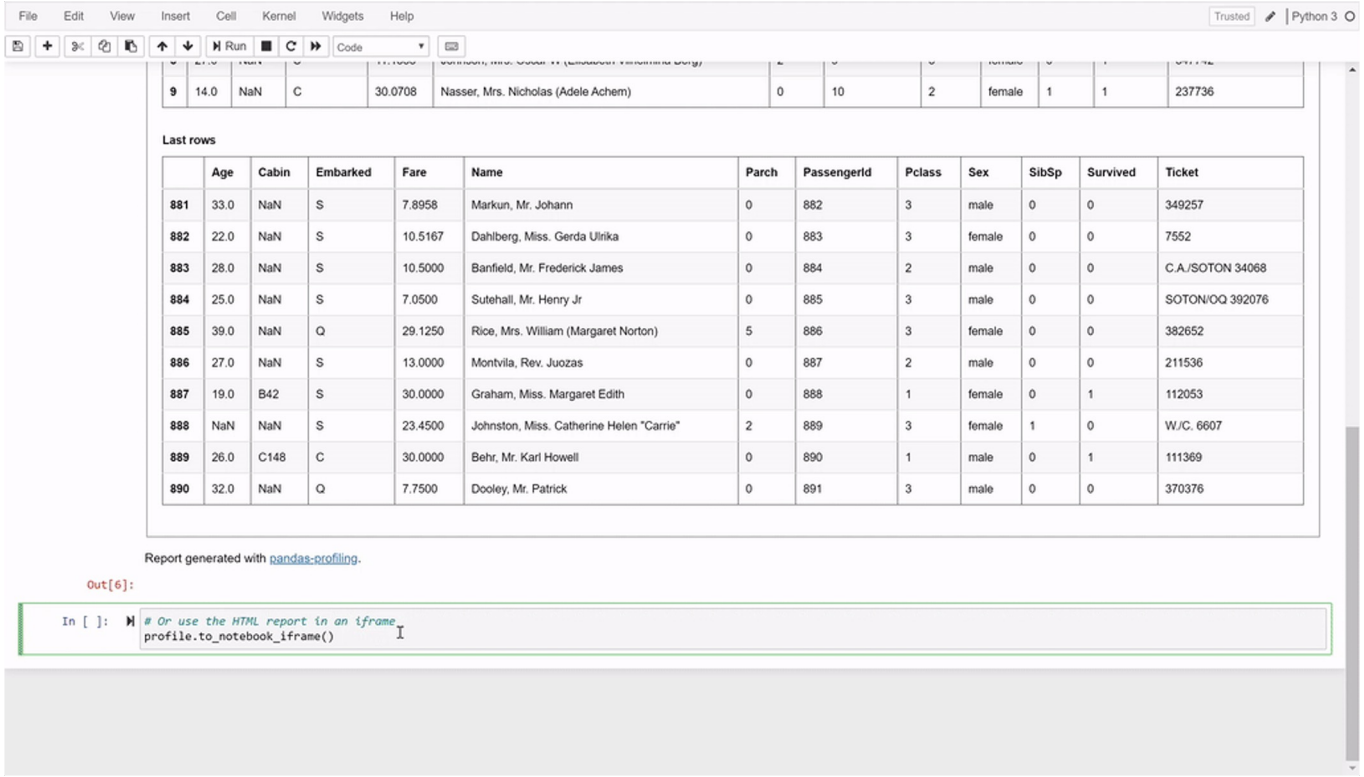

Slide from: CS 329 | Chip Huyen | cs329s.stanford.edu

Data profiling & visualization: Facets



https://pair-code.github.io/facets/

Data profiling & visualization: Pandas Profiling



https://github.com/pandas-profiling/pandas-profiling

Data assertions

- Common sense assertions
- Categorical data belongs to a predefined set
- Continuous data is within a range
- Outputs should follow a distribution
- Inputs should follow a distribution

Slide from: CS 329 | Chip Huyen | cs329s.stanford.edu

Data testing: specifying schema in your code

```python
from pydantic import BaseModel, ValidationError, validator


class UserModel(BaseModel):
    name: str
    username: str
    password1: str
    password2: str

    @validator('name')
    def name_must_contain_space(cls, v):
        if ' ' not in v:
            raise ValueError('must contain a space')
        return v.title()

    @validator('password2')
    def passwords_match(cls, v, values, **kwargs):
        if 'password1' in values and v != values['password1']:
            raise ValueError('passwords do not match')
        return v

    @validator('username')
    def username_alphanumeric(cls, v):
        assert v.isalnum(), 'must be alphanumeric'
        return v
```

```python
user = UserModel(
    name='samuel colvin',
    username='scolvin',
    password1='zxcvbn',
    password2='zxcvbn',
)
print(user)
#> name='Samuel Colvin' username='scolvin' password1='zxcvbn' password2='zxcvbn'

try:
    UserModel(
        name='samuel',
        username='scolvin',
        password1='zxcvbn',
        password2='zxcvbn2',
    )
except ValidationError as e:
    print(e)
    """
    2 validation errors for UserModel
    name
      must contain a space (type=value_error)
    password2
      passwords do not match (type=value_error)
    """
```


https://pydantic-docs.helpmanual.io/usage/validators/

Data testing: creating data schema

Table shape

- `expect_column_to_exist`
- `expect_table_columns_to_match_ordered_list`
- `expect_table_columns_to_match_set`
- `expect_table_row_count_to_be_between`
- `expect_table_row_count_to_equal`
- `expect_table_row_count_to_equal_other_table`

Missing values, unique values, and types

- `expect_column_values_to_be_unique`
- `expect_column_values_to_not_be_null`
- `expect_column_values_to_be_null`
- `expect_column_values_to_be_of_type`
- `expect_column_values_to_be_in_type_list`

```
expect_column_values_to_be_between(
    column="room_temp",
    min_value=60,
    max_value=75,
    mostly=.95
)
```


"Values in this column should be between 60 and 75, at least 95% of the time."

"Warning: more than 5% of values fell outside the specified range of 60 to 75."

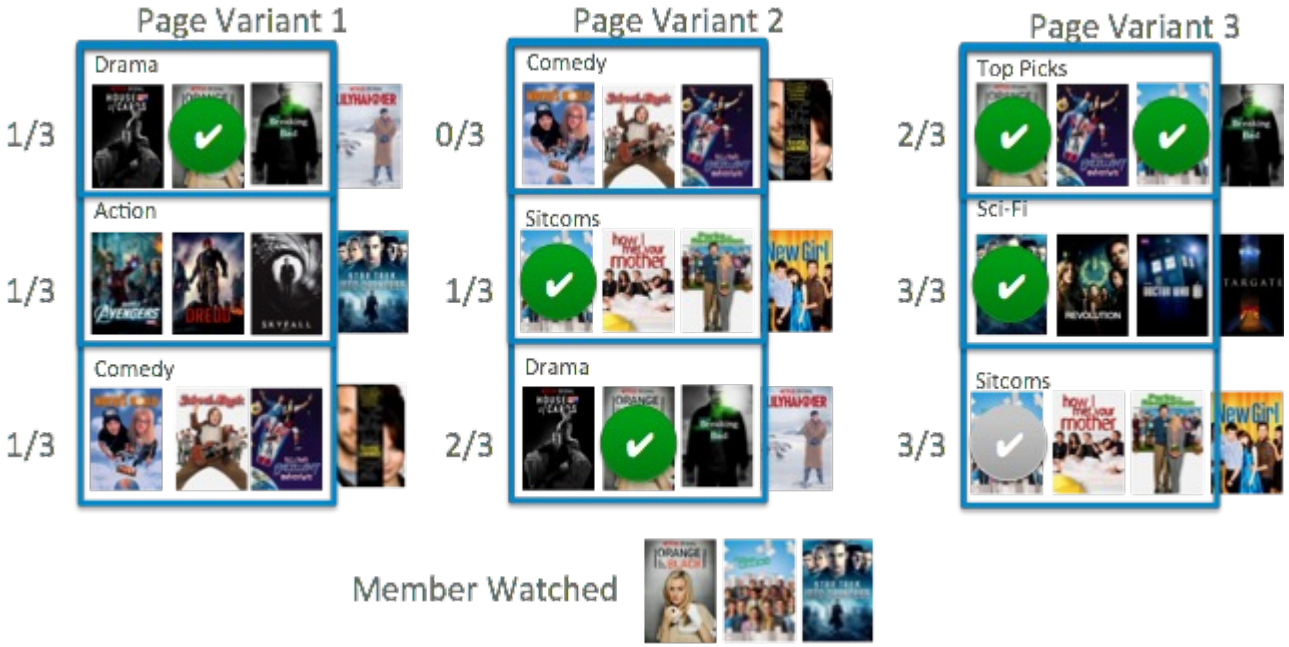
https://github.com/great-expectations/great_expectations

Netflix Page Generation Process

- They aim to create a scoring function by training a model using historical information of which homepages they have created for their members — including what they actually see, how they interacted with and what they played.
- There are many other features and ways that can represent a particular row in the homepage for the algorithm. It could be as simple as using all the item metadata (as an embedding) and aggregating them, or indexing them by position.
- Regardless of what features are used to represent the page, the main goal is to generate hypothetical pages and see what items the user would have interacted with.
- Scoring is then done using page-level metrics, such as *Precision@m-by-n* and *Recall@m-by-n* (which are adaptations of the *Precision@k* and *Recall@k* but in a two-dimensional space).

https://towardsdatascience.com/deep-dive-into-netflixs-recommender-system-341806ae3b48

Netflix Page Generation Process

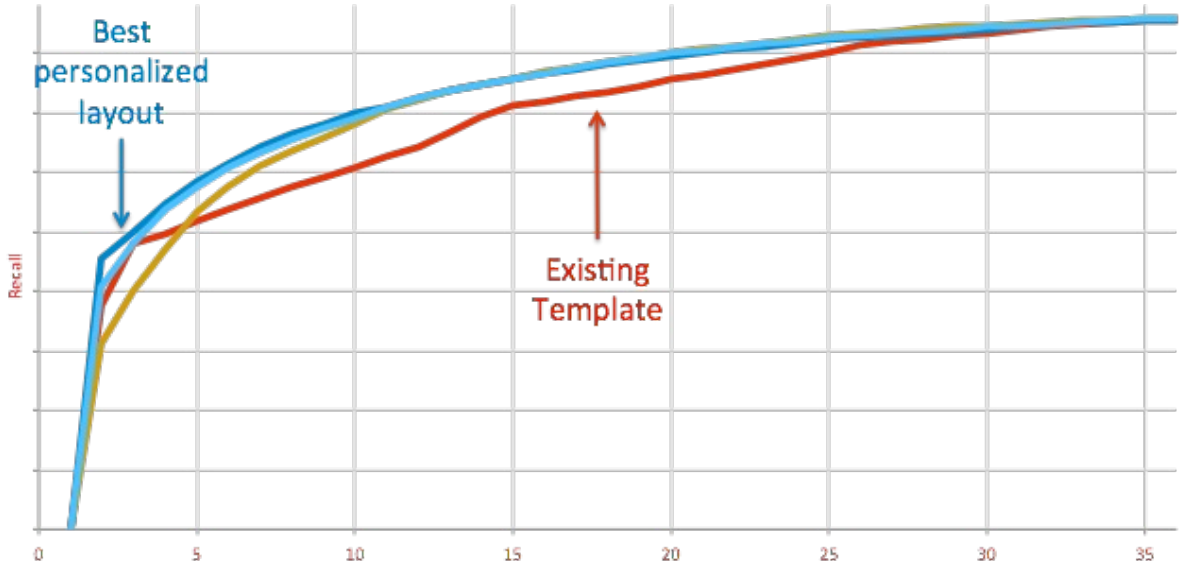


https://netflixtechblog.com/learning-a-personalized-homepage-aa8ec670359a

Netflix Page Generation Process

Comparison of four page algorithms in recall up to a fixed column position while sweeping the row position.

The red line is the previous rule-based approach and the blue is a personalized layout.



https://netflixtechblog.com/learning-a-personalized-homepage-aa8ec670359a

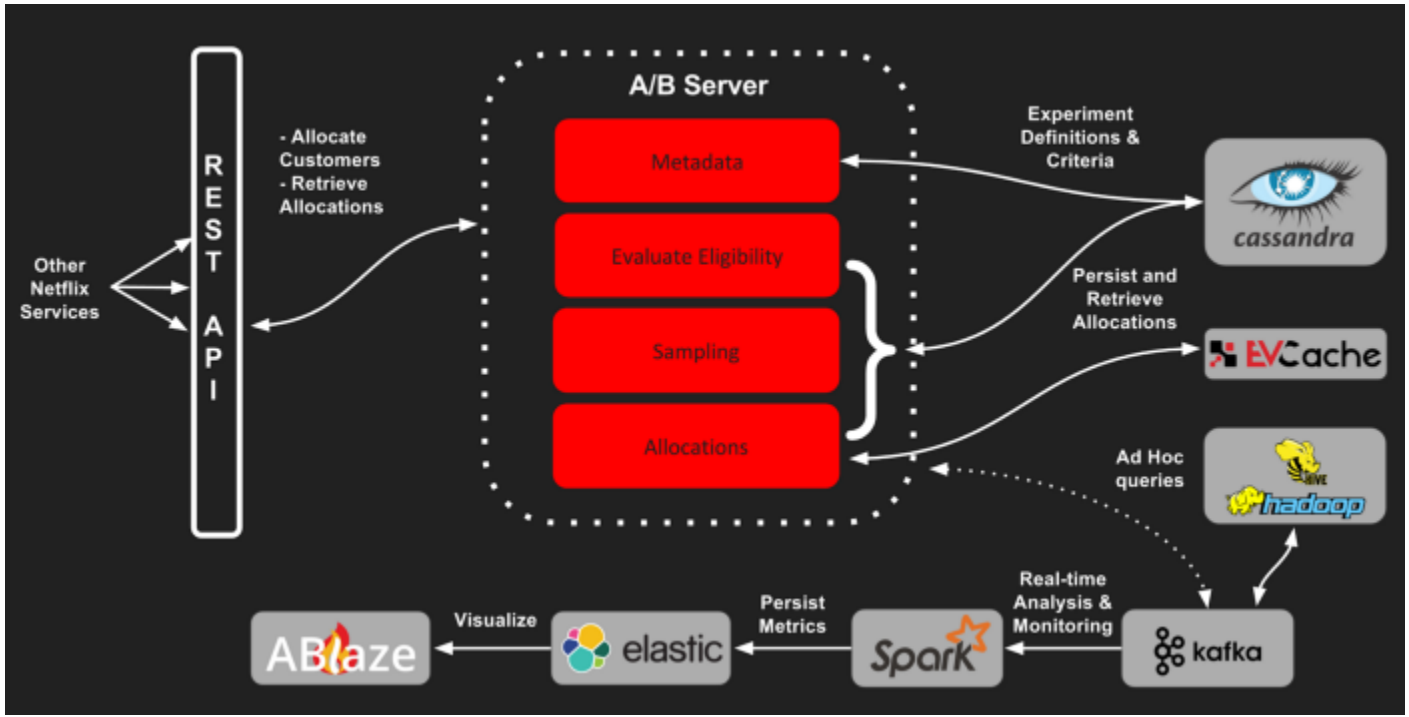
Other Challenges in Page Generation

There are challenging problems in engineering the homepage:

- When is it appropriate to take into account other context variables such as the time of the day or device, in how we populate the homepages?
- How do we find the appropriate trade-off between finding the optimal page and computational cost?
- How do we form the home pages during the critical first few sessions of a member, precisely at the time when we have the least information about them?

https://netflixtechblog.com/learning-a-personalized-homepage-aa8ec670359a

Netflix Testing



https://netflixtechblog.com/its-all-a-bout-testing-the-netflix-experimentation-platform-4e1ca458c15

Benchmarking

- Reproducible experiments with well-defined tasks, metrics, and rules
- Guide development
  - E.g. shared tasks on machine translation from 2006
- Measure progress
  - E.g. benchmark datasets/tasks for research
- Customers might require it (possibly to compare with other solutions)
  - E.g. latency, memory usage, prediction cost, accuracy of your model on their provided data
- Compete with other systems
  - E.g. MLPerf

Model evaluation

1. Slice-based evaluation
2. Perturbation evaluation
    a. Invariance tests
    b. Directional expectation tests
3. Ablation study

Slice-based evaluation

Models may perform differently on different slices:

- Classes
  - Might perform worse on minority classes
- Subgroups
  - Gender
  - Location
  - Time of using the app
  - etc.

Slice-based evaluation

Models perform the same on different slices with different cost

- User churn prediction
  - Paying users are more critical
- Predicting adverse drug reactions
  - Patients with underlying conditions are more critical

Simpson’s paradox

- Models A and B to predict whether a customer will buy your product
- A performs better than B overall
- B performs better than A on both female & male customers

Simpson’s paradox

	Treatment 1	Treatment 2
Group A	**93% (81/87)**	87% (234/270)
Group B	**73% (192/263)**	69% (55/80)
Overall	78% (273/350)	**83% (289/350)**

Numbers from a kidney stone treatment study. (Charig et al., 1986)

Simpson’s paradox: Berkeley graduate admission ‘73

	All		Men		Women	
	Applicants	Admitted	Applicants	Admitted	Applicants	Admitted
Total	12,763	41%	8442	**44%**	4321	35%

Sex Bias in Graduate Admissions: Data from Berkeley (Bickel et al., 1975)

Simpson's paradox: Berkeley graduate admission '73

Department	All		Men		Women	
	Applicants	Admitted	Applicants	Admitted	Applicants	Admitted
A	933	64%	***825***	62%	108	**82%**
B	585	63%	***560***	63%	25	**68%**
C	918	35%	325	**37%**	***593***	34%
D	792	34%	417	33%	375	**35%**
E	584	25%	191	**28%**	***393***	24%
F	714	6%	373	6%	341	**7%**

Aggregation can conceal and contradict the actual situation

Sex Bias in Graduate Admissions: Data from Berkeley (Bickel et al., 1975)

Slice-based evaluation

- Evaluate your model on different slices
  - E.g. when working with website traffic data, slice data among:
    - gender
    - mobile vs. desktop
    - browser
    - location
- Check for consistency over time
  - E.g. evaluate your model on data slices from each day

Slice-based evaluation

- Improve model's performance both overall and on critical data
- Help avoid biases
- Even when you don't think slices matter, slicing can:
  - give you confidence on your model (to convince your boss)
  - might reveal non-ML problems

How to identify slices?

- Heuristics
  - Might require subject matter expertise
- Error analysis
  - Patterns among misclassified samples
- Slice finder
  - Exhaustive/beam search
  - Clustering
  - Decision tree

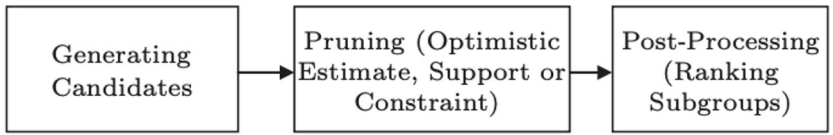


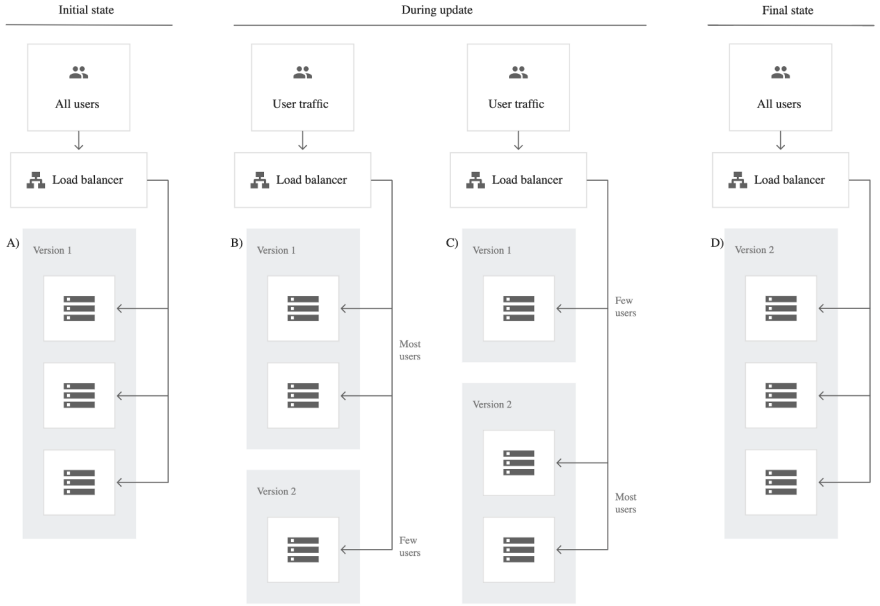
Fig.1. Methodology for subgroup discovery.

Slice finder: Automated data slicing for model validation (Chung et al., 2019)

Subgroup Discovery Algorithms: A Survey and Empirical Evaluation (Sumyea Helal, 2016)

Test in production
Canary testing

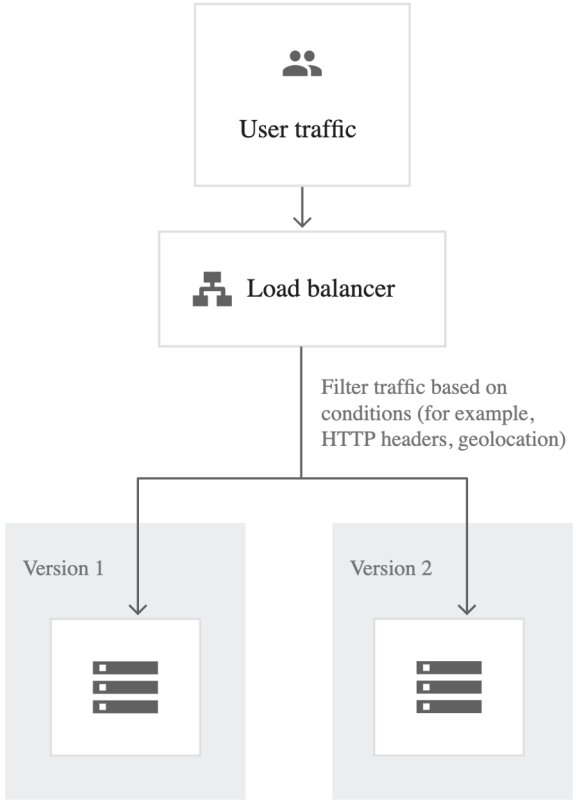
- New model alongside existing system
- Some traffic is routed to new model
- Slowly increase the traffic to new model
  - E.g. roll out to Vietnam first, then Asia, then rest of the world



Application deployment and testing strategies (Google)

A/B testing

- New model alongside existing system
- A percentage of traffic is routed to new model based on routing rules
- Control target audience & monitor any statistically significant differences in user behavior
- Can have more than 2 versions



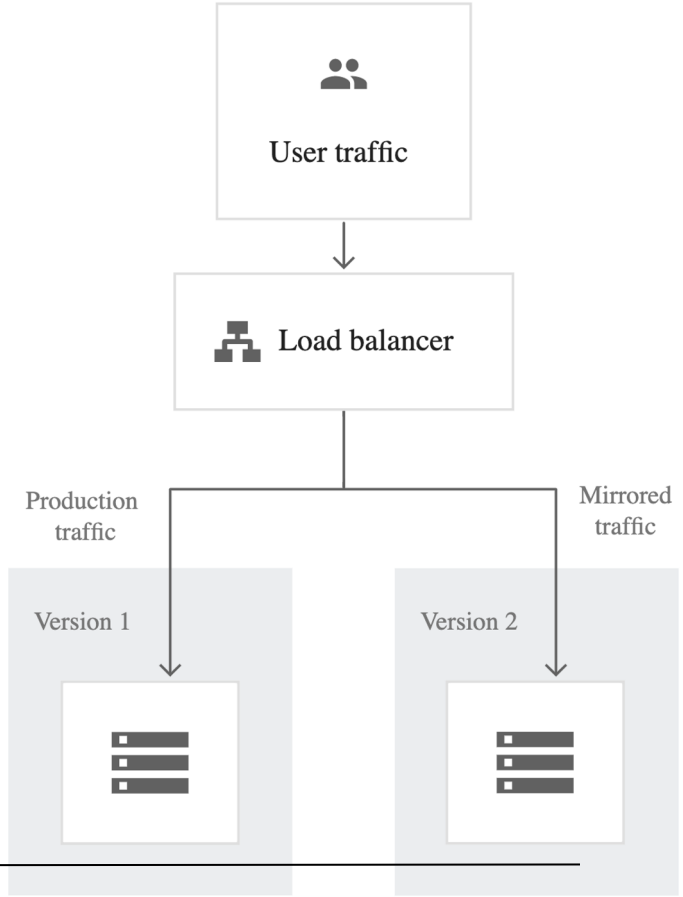
Application deployment and testing strategies (Google)

Interleaved experiments

- Especially useful for ranking/recsys
- Take recommendations from both model A & B
- Mix them together and show them to users
- See which recommendations are clicked on

Shadow testing

- New model in parallel with existing system
- New model's predictions are logged, but not show to users
- Switch to new model when results are satisfactory



Application deployment and testing strategies (Google)

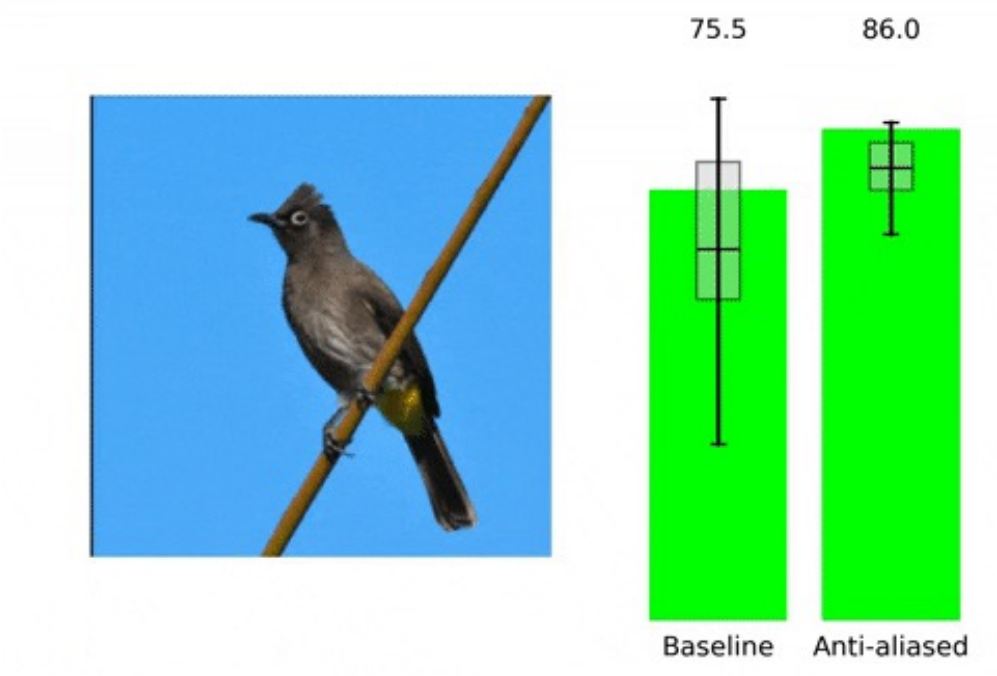
Test internally first

- Use features even as they’re in development
- Share internally before externally

Perturbation evaluation

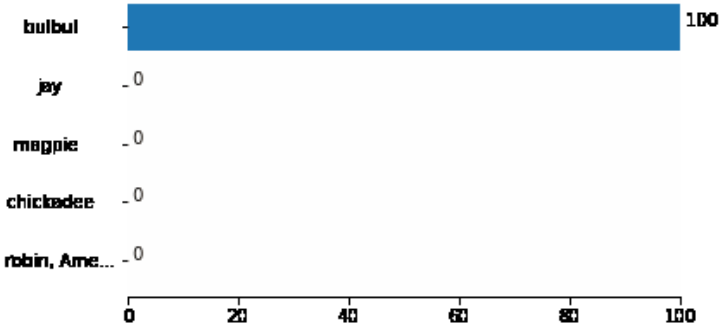
- Randomly add small noise to data to see how much outputs change
  - User inputs might contain noise
    - Speech recognition: background noise
    - Object detection: different lighting
    - Text inputs: typos, intentional changes (e.g. looooooooong)
  - Do small changes in inputs lead to big changes in outputs?

Small Shifts



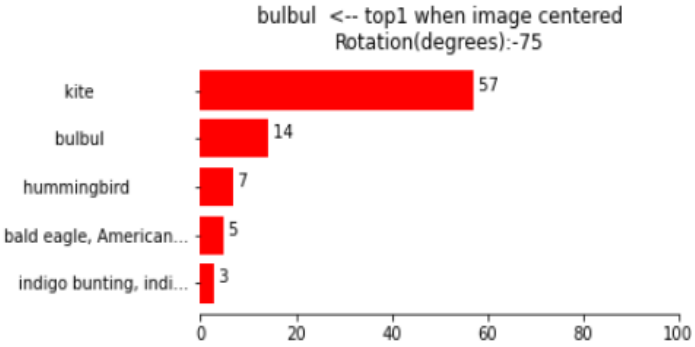
R. Zhang, Making Convolutional Networks Shift-Invariant Again, ICML, 2019.

Small Shifts



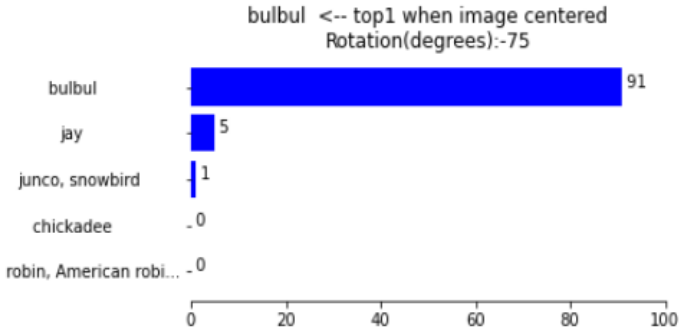
Resnet50 trained with random resize and crop augmentations

Image Rotation



Alexnet is sensitive against rotations: rotation may result in misclassification.

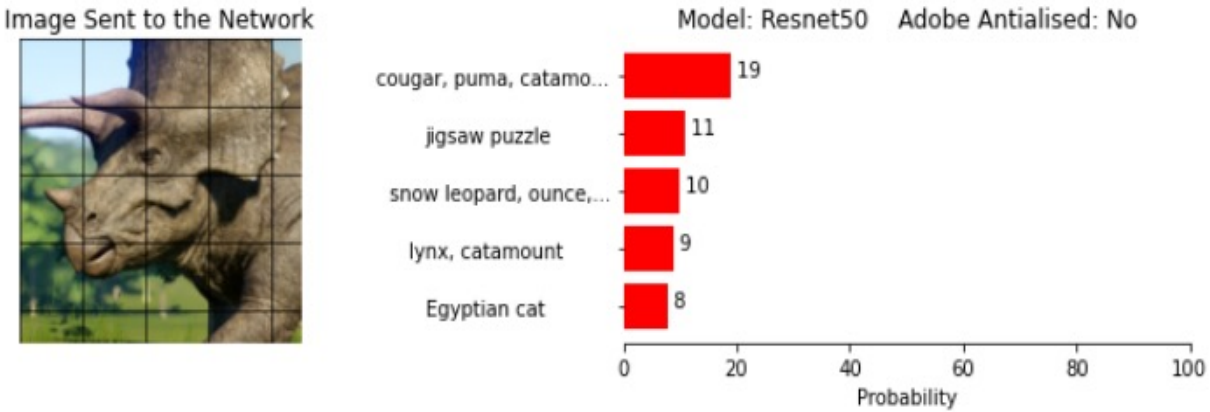
Image Rotation



On the other hand, relatively advanced networks like Resnet 18 are quite robust against rotation!

Adding a grid

- Adding a grid makes the image still easily identifiable to humans but may cause misclassification



Adversarial Attacks

Forcing a machine learning model to make wrong predictions by adding carefully crafted noises that are imperceivable to humans

Adversarial Perturbation: Noise added to a benign image

Adversarial Example: Perturbed version of an input image

Attack Example

"panda"

57.7% confidence

"gibbon"

99.3% confidence

Definitions

Black-box Attack: Adversary does not have knowledge about the target model (architecture, training procedure, etc.)

White-box Attack: Adversary has access to the model parameters

Non-targeted Attack: Adversarial example fools the model to make it predict any class than the correct one

Targeted Attack: Adversarial example is crafted to fool the model to make a predefined prediction

Fooling Rate: Percentage of how much the target model makes mispredictions on adversarial examples

Invariance tests

- Some input changes shouldn’t lead to changes in outputs
  - Changing race/gender info shouldn’t change predicted approval outcome
  - Changing name shouldn’t affect resume screening results

> The Berkeley study found that both face-to-face and online lenders rejected a total of 1.3 million creditworthy black and Latino applicants between 2008 and 2015. Researchers said they believe the applicants "would have been accepted had the applicant not been in these minority groups." That's because when they used the income and credit scores of the rejected applications but deleted the race identifiers, the mortgage application was accepted.

Disparity in home lending costs minorities millions, researchers find (CBS News, 2019)

Directional expectation tests

- Some changes to inputs should cause predictable changes in outputs
  - E.g. when predicting housing prices:
    - Increasing lot size shouldn’t decrease the predicted price
    - Decreasing square footage shouldn’t increase the predicted price

Ablation study

- Systematically removing parts of your model to see which ones are relevant to the model's performance
  - Features
  - Model components: e.g. regularizations

⚠️ Short-term results ≠ long-term results ⚠️

- Identical results short-term don’t mean identical results long-term
- Superior models now don’t mean superior models a week from now on

Netflix: Selecting the best artwork

- “If you don’t capture a member’s attention within 90 seconds, that member will likely lose interest and move onto another activity.”

- Netflix artworks are personalized based on your profile and preferences!

https://netflixtechblog.com/selecting-the-best-artwork-for-videos-through-a-b-testing-f6155c4595f6

Netflix: Selecting the best artwork

A/B tests Netflix did resulted in increased member engagement.

Their goals were the following:

- Identify artwork that enabled members to find a story they wanted to watch faster.
- Ensure that our members increase engagement with each title and also watch more in aggregate.
- Ensure that we don’t misrepresent titles as we evaluate multiple images.

https://netflixtechblog.com/selecting-the-best-artwork-for-videos-through-a-b-testing-f6155c4595f6

Netflix: Selecting the best artwork

- **Client side impression tracking:** One of the key components to measuring take rate is knowing how often a title image came into the viewport on the device (impressions). This meant that every major device platform needed to track every image that came into the viewport when a member stopped to consider it even for a fraction of a second. Every one of these micro-events is compacted and sent periodically as a part of the member session data. Every device should consistently measure impressions even though scrolling on an iPad is very different than the navigation on a TV. We collect billions of such impressions daily with low loss rate across every stage in the network — a low storage device might evict events before successfully sending them, lossiness on the network, etc.

https://netflixtechblog.com/selecting-the-best-artwork-for-videos-through-a-b-testing-f6155c4595f6