Week2 – Select and Train a model

Model Overview

\* Model-centric Vs Data-centric AI development

Key Challenges

\* AI System = Code + Data

\* Model+Hyperparameters+Data => ML Model

\* Model development is iterative

Challenges in Model development

1. Do well in training set
2. Do well in dev / test sets
3. Do well on business metrics/project goals

Why low average error isn’t good enough

1. Performance on disproportionately important examples
2. E.g. Informational and Transactional queries. Diwali -> Mistakes ok
3. Navigational queries -> youtube should get youtube.com. No mistake here
4. Performance on key slices of the dataset
   1. Do not discriminate, ML fairness
   2. Treat all user, retailer and product category fairly
5. Rare classes
   1. Skewed data distribution – 99% negative, 1%positive
   2. Accuracy in rare classes – Rare medical problems should still perform ok

Establish a baseline

1. Have human level performance and maybe that is a baseline
2. If humans did bad on low bandwidth no point improving that
3. Unstructured data – Images, audio, text. Human are good at interpreting. Human Level performance HLP is good baseline
4. Structured data – Data store in DB or excel sheets. Humans are bad at predicting here.
5. Literature search for state of the art/open source
6. Quick and dirty implementation
7. Performance of older system
8. Baseline helps to indicate what is possible. In some cases such as HLP it also gives a sense of what is irreducible error/Bayes error

Tips for getting started

1. ML is an iterative process, After a few iterations audit performance
2. Getting started on modeling – literature search
3. Find open-source implementations
4. A reasonable algo with good data.
5. Take deployment constraints when picking a model after baseline is established. Ignore until baseline constraints are established
6. Sanity check for code and algo. Try to overfit a small training dataset before training on a large one.

**Error Analysis and Performance Auditing**

**Error Analysis Example**

1. Find errors such as car noise, people noise, low bandwidth
2. Examine example tag Propose tags
3. Visual inspection
4. Product recommendations
5. User demographics
6. What fraction of errors has that tag?
7. Of all the data with tag, what fraction is misclassified
8. What fraction of all the data has that tag
9. How much room for improvement

**Prioritizing what to work on**

\* How much room for improvement there is

\* How frequently that category appears

\* How easy is it to improve accuracy in that category

\* How imp is it to improve in that category

\* Add improve data for specific categories

Collect more data

Use data augmentation

Use/Improved label accuracy and data quality

**Skewed datasets**

99.7% no defect, 0.3% defect, medical diagnosis

Raw accuracy is not useful

Confusion matrix – Precision and Recall are better

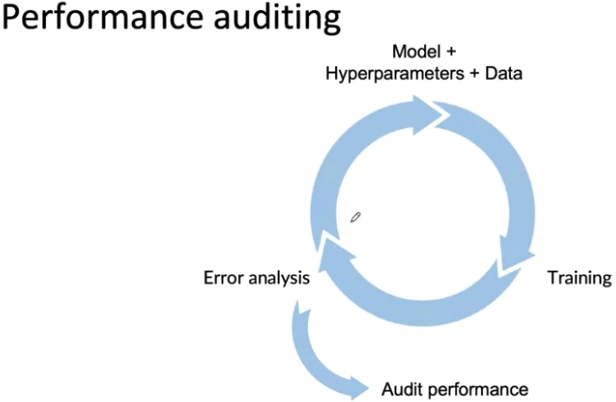
Precision = TP/TP+FP

Recall = TP/TP+FN

F1 score = Combine recall and precision. 2/1/p + 1/r = harmonic mean

Multi-class classification – Precision, recall, F1 useful here

**Performance auditing**



1. Check for fairness/bias, accuracy and other problems
2. Brainstorm the ways the system might go wrong.
3. Performance on subsets of data (ethnicity and gender)
4. How common are certain errors
5. Perf on rare classes
6. Establish metrics to assess perf against these issues on appropriate slices of data.
7. Get business/product owner buy in

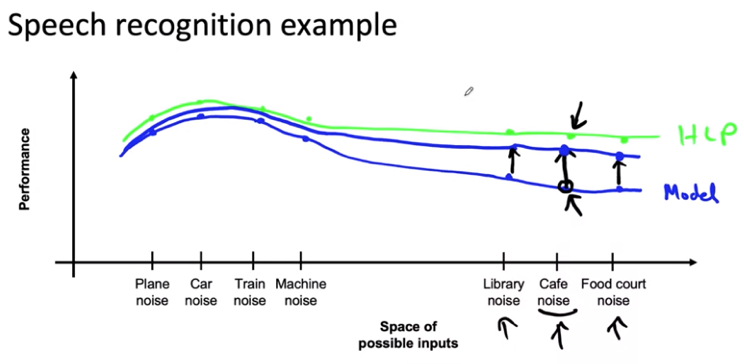
Data Iteration

**Data-centric AI development**

Model centric Keep improving the model. Hold data fixed improve the model

Quality of data is paramount. Use tools to improve data quality. This will allow models to do well. Fix model fixed and improve data

**A useful picture of data augmentation**



**Data augmentation**

1. Add noise to data
2. Create realistic examples that algo does poorly on, but humans do well on
3. Hard enough for algo but not hard enough to both algo and humans
4. Checklist
   1. Is data realistic
   2. Is x->y mapping clear
   3. Is algo doing poorly on this data
5. E.g. flip images, adjust contrast , adjust brightness, adjust lighting. Humans should be able to do well on.
6. Data iteration loop- Add improve data holding model fixed.

**Can adding data hurt?**

\* For unstructured data problems and if model is large and if mapping x->y is clear then adding data rarely hurts accuracy. X->Y mapping , humans should make accurate predictions.

e.g. Image, adding a lot of I’s may skew dataset and affect performance.

**Adding features**

For structured data, take existing data, and see if you can add new features to add

Restaurant recommendation example

\* Is person vegetarian

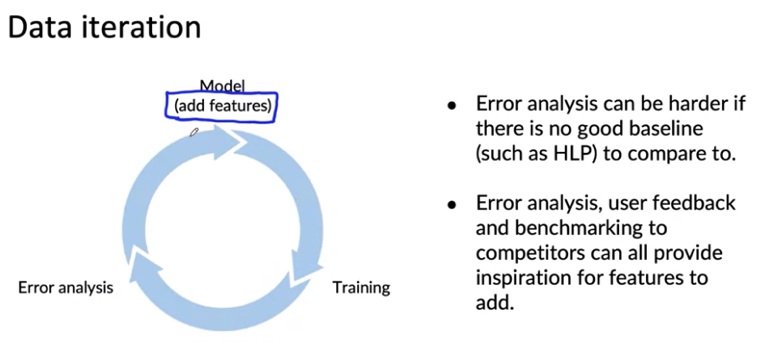
\* Does restaurant have vegetarian options

\* Human labeling

\* Shift from collaborative filtering to content based filtering

\* Advantage of content based filtering, look at the restaurant and make recommendations. COLD START PROBLEM

\* Data iteration Model-> Training -> Error Analysis-> model



**Experiment tracking**

\* What to track? What algo and code versioning

\* Track dataset use

\* Track hyperparameters

\* track results

Tracking tools

\* Experiment tracking system – Sage maker studio

Desirable features

\* Info needed to replicate results

\* Experiment results, with summary metrics and analysis

\* Resource monitoring, visualization, error analysis

**From big data to good data**

1. Good data
   1. Covers important cases
   2. Is defined consistently (definition of labels y is unambiguous)
   3. Has timely feedback from prod data (distrib covers data drift & concept drift)
   4. Is sized appropriately.

Additional Reading

[Establishing a baseline](https://blog.ml.cmu.edu/2020/08/31/3-baselines/)

[Error analysis](https://techcommunity.microsoft.com/t5/azure-ai/responsible-machine-learning-with-error-analysis/ba-p/2141774)

[Experiment tracking](https://neptune.ai/blog/ml-experiment-tracking)

Papers

Brundage, M., Avin, S., Wang, J., Belfield, H., Krueger, G., Hadfield, G., … Anderljung, M. (n.d.). Toward trustworthy AI development: Mechanisms for supporting verifiable claims∗. Retrieved May 7, 2021<http://arxiv.org/abs/2004.07213v2>

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., & Sutskever, I. (2019). Deep double descent: Where bigger models and more data hurt. Retrieved from <http://arxiv.org/abs/1912.02292>