ORES Custom Documentation V

Disclaimer: No guarantee for the correctness of information / explanations / sources is given.

Goals

1. Metrics list:
   - Add examples ✓
   - Correct the descriptions of counts and rates ✓
   - Improve descriptions of roc_auc and pr_auc ✓
   - Add the new standalone version to the repo ✓
2. Research what form of revision data is needed (for existing visualizations, but also in general) → also check RecentChanges and new Filters and what API calls look like in that context
3. - Watch "ROC curves and Area Under the Curve explained" and add insights to the **roc_auc** in the list ✓
   - and think about what parameters could be used in which ways to filter the output of the current UI (currently: X inputs → X outputs)
   - Also check out: Precision-Recall AUC vs ROC AUC discussion ✓
4. Read A Review of User Interface Design for Interactive Machine Learning
5. Think about what could be the goal of this thesis

1 Crucial metrics: damaging-model

Examined metrics:

- !f1
- !precision
- !recall
- accuracy
- counts
- f1
- filter_rate
- fpr
- match_rate
- pr_auc
- precision
- rates
- recall
- roc_auc

For each metric (if possible) there will be:

1. The formula based on the **confusion matrix**
2. A definition
3. An intuitive explanation with an example
4. Its meaning based on the **loan threshold** representation by Google (Link)
5. Additional information (if necessary)

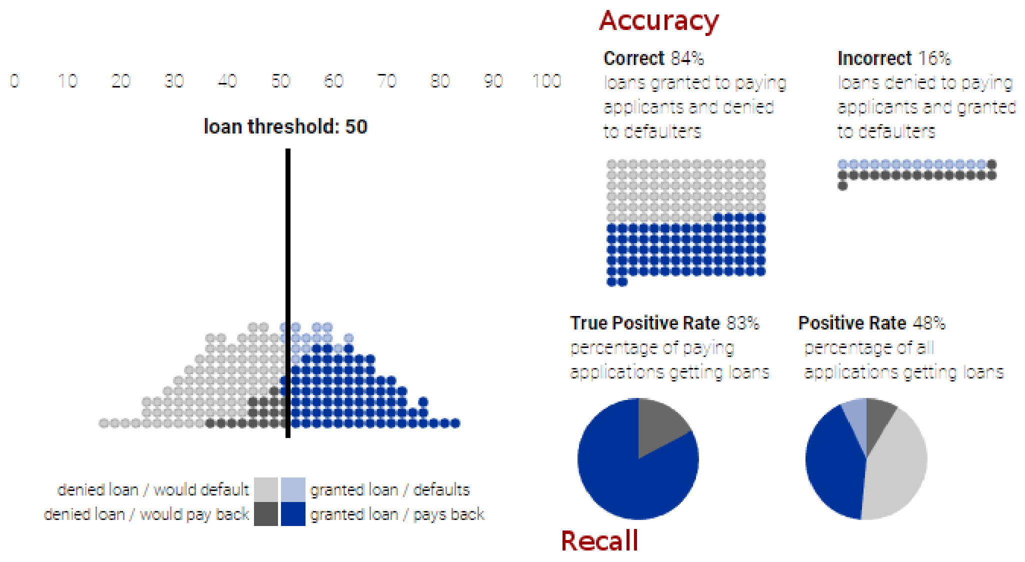
Explanations: References

- Confusion Matrix

		Actual	
		Positive	Negative
Predicted	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

Abbreviations: **TP**, **FP**, **FN** and **TN**.

- Loan Threshold

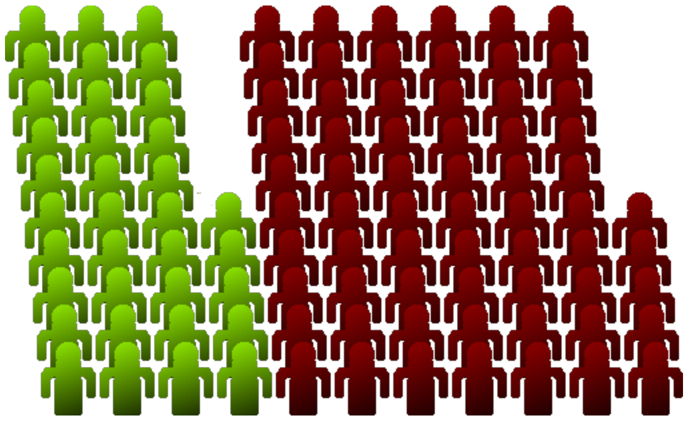


The example scenario

To ease the understanding, let's stick to the following scenario and refer to it for each metric:

- 100 people represent our total population
- 35% of our population is infected with disease X

  That leaves us with the following **labels**: **35 positives** and **65 negatives** (being tested for disease X;)



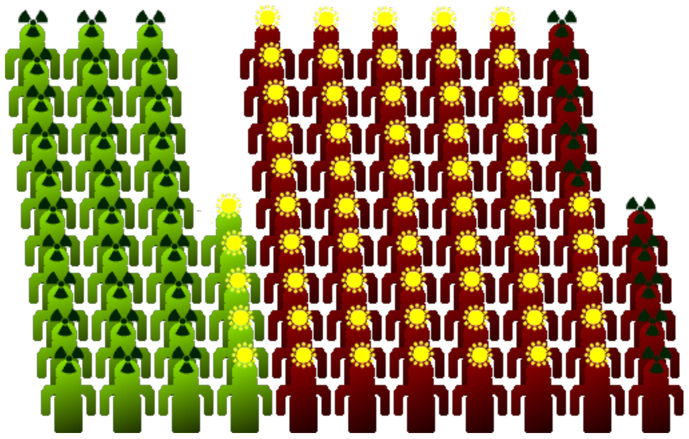
- A classifier is now supposed to classify every person of our population (based on their visible symptoms for example). This is the **prediction**:
  - If our algorithm says a person is infected, that person will be classified as a **positive**, and marked with radioactive symbol:

    ☢

  - If the prediction results in a **negative**, the person will be marked with a sun symbol:

    ☀

- The classifier may predict that:
  - out of the 35 infected people, **30** are infected (those 30 are what we call **true positives**) and **5** are not (those 5 are **false negatives**)
  - out of the 65 non infected people, **10** are infected (**false positives**) and **55** are not (**true negatives**)



true positive: infected and correctly predicted (**30**)

false negative: infected and incorrectly predicted (**5**)

true negative: not infected and correctly predicted (**55**)

false positive: not infected and incorrectly predicted (**10**)

Let's get started.

1.1 recall

1. $\frac{\texttt{TP}}{\texttt{TP+FN}}$

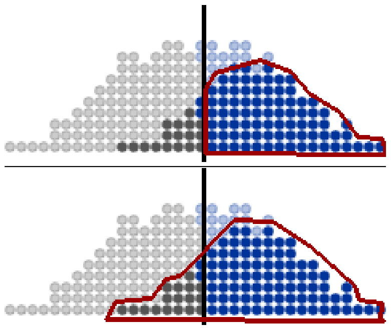
2. Recall ($\equiv$ true positive rate $\equiv$ "sensitivity") is the ability of a model to find **all** relevant cases within the dataset.

3. Now, in our example scenario, the relevant cases are the infected people. We absolutely want to identify those: . The ability of the model to identify those depends on how many will be **correctly** predicted to be infected: .

   In other words, we are looking for the ratio of correctly predicted to be infected people to all infected people.

That leads to $\frac{\sum 🟩}{\sum 🟩}$, with 🟩 = 🟩 + 🟩, which is equivalent to the formula in **1.**, if you replace the symbols with their confusion matrix counterpart according to the legend in **The example scenario**.

In terms of numbers for our example that would be $\frac{30}{30+5} \approx 0.86$

4. 

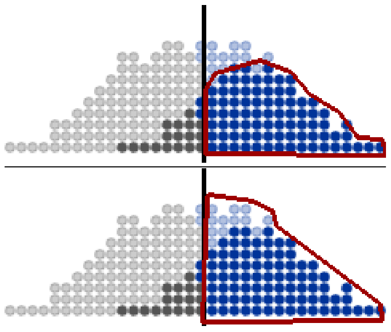
1.2 precision

1. $\frac{\mathtt{TP}}{\mathtt{TP}+\mathtt{FP}}$

2. Ability of the model to find **only** relevant cases within the dataset

3. Again, we take a look at the relevant cases, the infected people: 🟩. This time around though, we are **not** interested in the ratio of correctly predicted to be infected people to **all** infected people. Instead we want to know how good the model is at only predicting those to be infected, that actually are. Therefore, we want the ratio of all 🟩 to all those predicted to be infected: 🟩 + 🟥

   $$= \frac{\sum 🟩}{\sum 🟩 + \sum 🟥} = \frac{30}{30+10} = 0.75$$

4. 

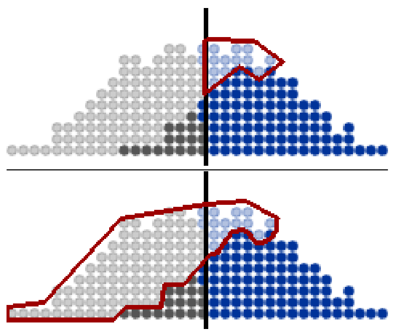
1.3 f1

1. -
2. f1-Score, the harmonic mean of recall and precision, a metric from **0** (worst) to **1** (best), used to evaluate the accuracy of a model by taking into account recall and precision: $= 2 * \frac{\texttt{precision} * \texttt{recall}}{\texttt{precision} + \texttt{recall}}$
3. For our example model, that would result in $= 2 * \frac{0.75 * \frac{30}{35}}{0.75 + \frac{30}{35}} = 0.8$
4. -
5. Additional information: Compared to the simple average (of recall and precision), the harmonic mean punishes extreme values (e.g. precision 1.0 and recall 0.0 $\rightarrow$ average = 0.5, but f1 = 0)

1.4 fpr

1. $\frac{\texttt{FP}}{\texttt{FP}+\texttt{TN}}$
2. The false positive rate is the probability of a false alarm.
3. In our example, a false alarm would obviously be labeling someone as infected, who isn't: . Now we just have to ask ourselves what portion of those, that, if they **were** incorrectly predicted as infected (because they **are not** infected:), **are** incorrectly predicted as infected? Hence, we are looking for the ratio of to all non infected people: $+$.

   $$= \frac{\sum}{\sum + \sum} = \frac{10}{10+55} \approx 0.15$$

4. 

1.5 roc_auc

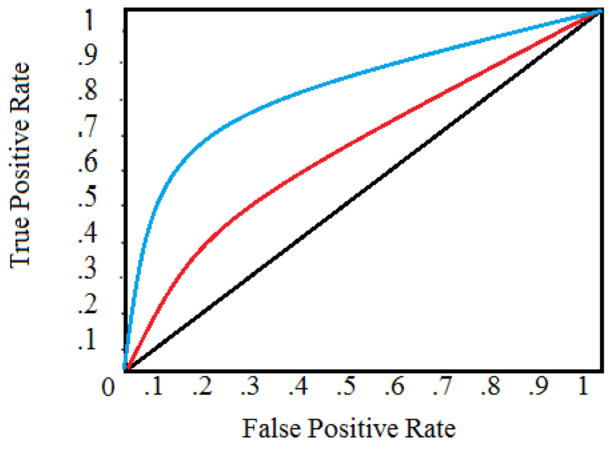
1. -

2. The **area under** the **curve** of the **ROC**-curve, a measure between 0.5 (worthless) and 1.0 (perfect: getting no FPs), rates the ability of a model to achieve a blend of recall and precision.

3. In our example, we haven't used the notion of threshold yet. For classifying people as infected or not, the classifier will evaluate multiple criteria and calculate the probability that a patient is infected. Many binary classifiers have the threshold at 0.5, meaning that, if the probability of a true outcome is higher than 50%, it is classified as a positive; or in our case as an infected patient. Depending on the situation however, it can be useful to move that threshold.

   The receiver operating characteristic (ROC) curve is used to visualize the performance of a classifier

   The ROC curve plots the TPR versus FPR as a function of the model's threshold for classifying a positive.



   Increasing the threshold $\rightarrow$ moving up a curve ($\equiv$ model) to the top right corner, where all data is predicted as positive (threshold = 1.0) and vice versa

   Our best case is a curve hugging the top left corner, because that means a low **fpr** and high **tpr**, which, again, means a lot of positives and few negatives on the right of the thresholds, looking at class curves like the **Loan Threshold** one.

Assuming that we have had a threshold of 0.5 all along to get the previous results, one point on our ROC curve would be: $(0.15, 0.86) = (\texttt{fpr}, \texttt{tpr})$.

We would now have to change the thresholds, look at the (probably) changed resulting data (new numbers in terms of positives and negatives, as well as **TP**, **FN**, **TN** and **FP**) and then plot every resulting $(\texttt{fpr}, \texttt{tpr})$-point in order to plot the full ROC curve.

We would most certainly like to quantify this visualized performance of our binary classifier, which is why we calculate the area under the curve (**auc**) of the ROC curve.

4. -

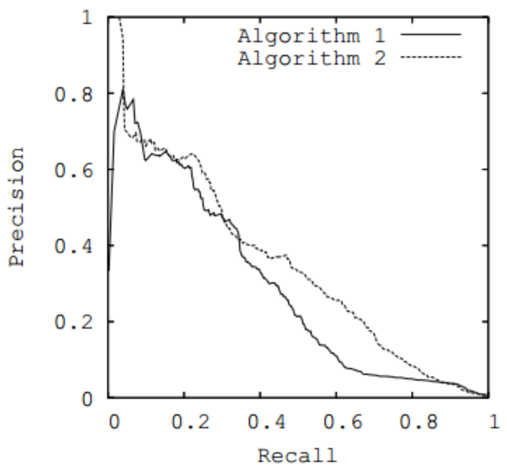
5. Additional information: We can think of AUC as representing the probability that a classifier will rank a randomly chosen positive observation higher than a randomly chosen negative observation. That's why **roc_auc** is a useful metric even for datasets with highly unbalanced classes. (Source)

1.6 pr_auc

1. -

2. Similarly to the **roc_auc**, the area under the precision recall curve **pr_auc** evaluates a classifiers performance. The main difference, however, is that the **pr_auc** does not make use of **true negatives**. It is therefore favourable to use **pr_auc** over **roc_auc** if true negatives are unimportant to the general problem or if there are a lot more negatives than positives.

   The PR-curve plots the Precision versus the Recall:



Instead of the top left corner for the ROC-curve, here, we want our curve to reach the top right corner for our classifier to be perfect.

The following scenario provides a good example (Source 1 and Source 2) of a case with a lot more negatives than positives and comparing ROC to PR:

- Out of 1 million documents, we want to find the 100 relevant ones.
- The task is accomplished by two different algorithms:
  (a) 100 retrieved documents, 90 relevant
  (b) 2000 retrieved documents, 90 relevant
- Algorithm (a) is obviously preferable.
- We know that ROC- and PR-curves both plot coordinates with `tpr` = `recall` as one dimension. Now the question is: how do they differ in the other dimension, when plotting both algorithms?
- In all cases `tpr` = `recall` = 0.9. We also have:
  (a) $\texttt{TN} = 999890$ and $\texttt{FP} = 10$
  (b) $\texttt{TN} = 997990$ and $\texttt{FP} = 1910$
- ROC:
  (a) $\texttt{fpr} = \frac{\texttt{FP}}{\texttt{FP+TN}} = \frac{10}{10+999890} = 0.00001$
  (b) $\texttt{fpr} = \frac{1910}{1910+997990} = 0.00191$

  Having retrieved many more documents, and therefore having many more **false positives**, algorithm (b) has a higher **fpr** than algorithm (a).

The **fpr** also takes into account the vast amount of **true negatives** though, which is why the difference between the two **fpr**s is still quite small: 0.0019.

- PR:

  (a) $\texttt{precision} = \frac{\texttt{TP}}{\texttt{TP+FP}} = \frac{90}{90+10} = 0.9$

  (b) $\texttt{precision} = \frac{90}{90+1910} = 0.045$

  Not accounting for **true negatives**, the **precision** is not affected by the relative imbalance.

  We are presented a remarkable difference of 0.855.

- To close on this topic, not the following (by Randy C): "Obviously, those are just single points in ROC and PR space, but if these differences persist across various scoring thresholds, using ROC AUC, we'd see a very small difference between the two algorithms, whereas PR AUC would show quite a large difference."

3. Similarly to the **roc_auc**, the point on the PR curve of our example for the standard threshold of 0.5 would be: $(\texttt{precision}, \texttt{recall}) = (0.75.0.86)$.

4. -

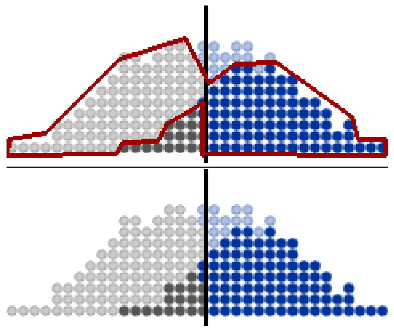
1.7 accuracy

1. $\frac{\texttt{TP+TN}}{\texttt{Total}}$

2. Accuracy measures the portion of correctly predicted data

3. In our example scenario, this is equal to asking ourselves *out of all patients, what's the portion of correctly predicted cases?* The correctly predicted cases are infected patients, predicted to be infected ( ), and non infected patients, predicted not to be infected ( ). This wanted proportion results in:

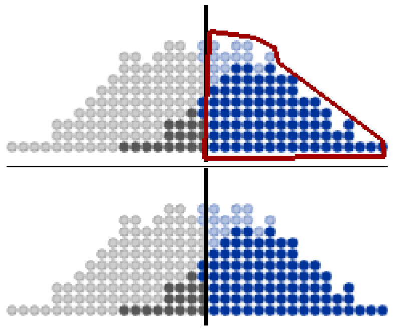
$$= \frac{\sum + \sum}{\sum + \sum} = \frac{30+55}{35+65} = 0.85$$

4. 

1.8 match_rate

1. $\frac{\texttt{TP+FP}}{\texttt{Total}}$
2. The match rate is the proportion of observations matched/not-matched, meaning the ratio of observations predicted to be positive to the total of observations.
3. Concerning our example, this would be equal to wanting to know what portion of the population was predicted to be infected. Those groups are: 🟩 and 🟥.

$$= \frac{\sum \text{🟩} + \sum \text{🟥}}{\sum \text{🟩} + \sum \text{🟥}} = \frac{30+10}{35+65} = 0.4$$

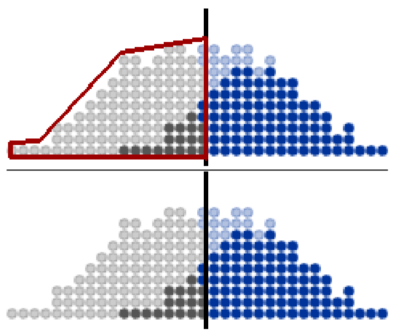
4. 

1.9 filter_rate

1. $1 - \texttt{match\_rate} = \frac{\texttt{TN+FN}}{\texttt{Total}}$
2. The filter rate is the proportion of observations filtered/not-filtered, meaning the ratio of observations predicted to be negative to the total of observations. This is the complement to the match rate.

3. In our example scenario, this would be equal to wanting to know what portion of the population was predicted not to be infected.

$$= \frac{\sum + \sum}{\sum + \sum} = \frac{55+5}{35+65} = 0.6 = 1 - \texttt{match\_rate}$$

4. 

1.10 counts

1. - **labels**:
     - false: $\texttt{TN} + \texttt{FP}$
     - true: $\texttt{TP} + \texttt{FN}$
   - **n**: `Total`
   - **predictions**:
     - false:
       - false: `TN`
       - true: `FP`
     - true:
       - false: `FN`
       - true: `TP`

2. - **labels**: The number of edits (*manually*) labeled as *false* and *true*: these values represent the **actual** positives and negatives.
   - **n**: The sample size; total number of edits taken into account
   - **predictions**: edits ...
     - false: ... actually being false ...
       - false: ... and predicted to be false
       - true: ... but predicted to be true
     - true: ... actually being true ...

        * false: ... but predicted to be false
        * true: ... and predicted to be true

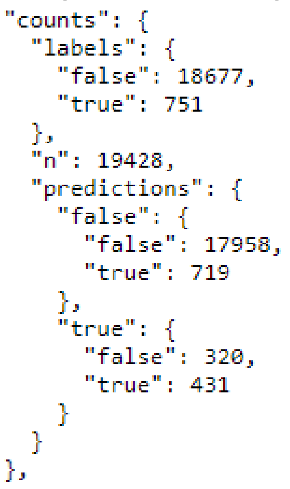
3. Concerning our example:

    - **labels**:
        - false (non infected people): + = 55 + 10 = 65
        - true (infected people): + = 30 + 5 = 35
    - **n** (Total): 100
    - **predictions**:
        - false (non infected people ...)
            * false (... and predicted not to be infected): = 55
            * true (... but predicted to be infected): = 10
        - true (infected people ...)
            * false (... predicted not to be infected): = 5
            * true (... predicted to be infected): = 30

4. -

5. Additional information:

    When calling the enwiki damaging model for example (Link), you will get the following output for counts:



```
"counts": {
  "labels": {
    "false": 18677,
    "true": 751
  },
  "n": 19428,
  "predictions": {
    "false": {
      "false": 17958,
      "true": 719
    },
    "true": {
      "false": 320,
      "true": 431
    }
  }
},
```

$\Rightarrow$ e.g. out of 18677 edits that were labeled as *false*, 719 were false positives

1.11 rates

1. - false: $\frac{\texttt{counts\_labels\_false}}{\texttt{counts\_n}}$
   - true: $\frac{\texttt{counts\_labels\_true}}{\texttt{counts\_n}}$

2. The rates simply give us the proportion of edits labeled as false or true to the total number of edits taken into account.

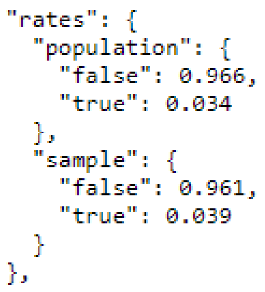
3. For the example scenario:

   - false (proportion of infected people to the total number of people tested): $\frac{\sum \text{🟥}}{100} = \frac{65}{100} = 0.65$
   - true (proportion of infected people to the total number of people tested): $\frac{\sum \text{🟩}}{100} = \frac{35}{100} = 0.35$

4. -

5. Additional information:

   Calling the API the same way as for **counts** (Link), we get:



```
"rates": {
  "population": {
    "false": 0.966,
    "true": 0.034
  },
  "sample": {
    "false": 0.961,
    "true": 0.039
  }
},
```

   The number of edits taken into account for **sample** equals the **n** from **counts** = 19428.

   Now we see that, also looking at the output under **counts**, *rates_sample_false*: $0.961 = \frac{18677}{19428}$.

   Note that we are shown results for "population" and "sample". There is a significant number of bot edits and edits that don't need reviewing (admins, autopatrolled users). The **sample** of edits does not contain any of those.

1.12 !<metric>

- Any <metric> with an exclamation mark is the same metric for the negative class
- e.g. recall = $\frac{\texttt{TP}}{\texttt{TP+FN}} \Rightarrow$ **!**recall = $\frac{\texttt{TN}}{\texttt{TN+FP}}$
- Example usage: find all items that are not “E” class (itemquality model) $\rightarrow$ look at **!**recall for “E” class.

1.12.1 Existing !<metric>s

- !f1
- !precision
- !recall

2 Revision data, RecentChanges and newFilters

3 pr_auc, roc_auc and filtering the ORES UI output

3.1 pr_auc and roc_auc

Insights from the linked video and discussion under **Goals** have been added to the metrics list.

3.2 What parameters could be used to filter the output of the current UI?

4 A Review of User Interface Design for Interactive Machine Learning - Notes

- Interactive Machine Learning (**IML**) complements human percept. + int.: computat. speed + power
- Interactive process: input from user without needing knowledge of ML

  $\Rightarrow$ UI Design is fundamental!

... but no consolidated principles on implement. of such a UI

- In this paper: proposit. of model of generalised IML system and solut. principles for effective IML interf.

4.1 Introduction

- ML: application as development tool only for experts
- IML: training process as HCI task $\rightarrow$ more accessible

  for example: human input in example selection, creation and labelling
- Note: expert may still be required to underlying alg.
- Ideal case: non expert constructs own learned concepts by creating or collecting training data according to their need

  In practice: major challenge for ML practitioner (=expert) and UI designer!

  $\Rightarrow$ This paper: focuses on interface design challenge
- IML: allows for application of domain knowledge (domain experts can now train models)!
- IML workflow is co-adaptive: user and target model directly influence each other's behaviour

  Again, this is done over the interface: it's a dialogue between human and machine
- Four key challenges for designing and IML interface:

  1. Users are imprecise and inconsistent. "Unless users perceive their own deficiencies, this failure [poor model resulting from poor training by user] is attributed to the system."
  2. Uncertainty relating user input and user intent. E.g. user doesn't assign an example to a concept does not imply a counter example
  3. Model is not your conventional information structure: "ML model evolves in response to user input but not necessarily in a way that is perceived as intuitive or predictable by the user"
  4. Training is open ended.

4.1.1 Guidance for the designer

Structural and behavioural breakdown of IML system will be presented.

Help designers think about the architecture and support discussion of design considerations with clear terminology

Also help with how to present information to users and what interactions to promote

- Structural: Constituent components (user, interface, data, model)

  Interface will be divided into four elements itself.

- Behavioural: Seperate process of interactively building model into subtasks

  Tasks may overlap / be performed iteratively

4.1.2 Outline

- Section 2: Detailed definition of IML
- Section 3: Overview of efforts at making ML more accessible
- Section 4: Key interface elements of generalised IML stystems
- Section 5: IML process: an abstracted description
- Section 6: Identifying common solution principles
- Section 7: Proposition of further research
- Section 8: Summary

4.2 Context and definitions ("Section 2")

4.2.1 IML

- *Def.* : Interact. paradigm where user (+-group) iteratively builds and refines mathematical model to describe concept through iterative cycles of input and review

- Differences to traditional ML:

  Human int. is applied through iterative teaching in tight loops

$\Rightarrow$ change in the model at each iterat. is relatively small

But also: less pre-selection of training data needed

User is principle driver of interaction

- Examples of IML: Crayons and ReGroup. IML is esp. beneficial when objectives of user "and/or data labels cannot be obtained a priori"
- **User(s)** *def.* :
  - Main driver
  - No underst. of ML
  - Possibly significant domain expertise
  - Multiple users may working in parallel
  - Dynamic and unreliable
- **Model** *def.* :
  - Takes inputs
  - Determines outputs based on current understanding of the concept
  - May be manipulated directly (changing parameters) or indirectly (relabelling/adding data)
  - May be unreliable too (non deterministic)
- **Data** *def.* :
  - Selected, described, generated by user
  - Indicates how model should behave to inputs
- **Interface** *def.* :
  - Central focus of this paper
  - Enables bidirectional feedback betw user and model/data

4.2.2 Related domains

- MLaaS: ML as a Service; ML platforms by Google, Amazon etc. but still require familiarity with techniques
- Recommender systems: influenced by (dis)likes, buying etc. $\leftarrow$ user often unaware and often focus on data capture instead of closed-loop model refinement

- Active learning: main difference to IML is focus on selecting new points for labelling by user to an extend where we can't talk about user driven input anymore
- Reinforcement learning
- Human-robot interaction research

"IML differs from these fields in the emphasis given to the role of the user and the process rather than the end goal."

4.3 Survey of IML Applications

Survey of existing approaches categorised by data type

4.3.1 Text

Text is often available in high volumes, often already has some sort of labelling

- **abstrackr**: "online tool to support screening for systematic reviews of published medical evidence" (classification of relevant citations) by Wallace et al.
- IML system for statutory analysis (by Savelka et al): facilitate the process of reviewing regulatory frameworks (public health)
- Classifying student code sumbissions: interactive interface by Kim et al.
- See paper for more examples by Yimam et al., Huang et al., Endert et al.

4.3.2 Images

Already a very popular application area for ML in general $\Rightarrow$ availability of suitable techniques

- *CueFlik* (Fogarty et al.): "interactive application that allows users to create their own concepts to incorporate into image search"

  Interesting: "providing a visualisation of both the most-confident positive and most-confident negative samples led to higher quality and faster concept learning than providing the full ranking of images"

- Clustering medical images: IML system by Guo et al.
- Again, see paper for more examples

Also note: successful interaction technique is (amongst others) "provid[ing] pixel or region level feedback by directly sketching on images"

4.3.3 Time series data: speech, audio, video and motion

- Automatic speech recognizer + IML interface for error correction (Sanchez-Cortina et al.)
- Automatic recognition of animal behaviours (video footage) by Kabra et al.
- Filtering surveillance streams for safeguarding nuclear processing facility: help identify potential hazards (Versino and Lombardi)
- Seperation of individual sources from recorded audio track with "paint-on" approach (on regions of time-frequency visualisation of audio recording by Bryan et al.
- *BeatBox*: users train system to recognise different percussive vocalisations (Hipke at al.)
- See paper for more examples

Interesting: "IML is not necessarily just about making a task more efficient or productive, but potentially about extending human creativity"

In regards to time series data, interfaces "must find ways to allow users to quickly inspect and review samples" $\leftarrow$ major challenge

4.3.4 Assisted processing of structured information

Structured data $\rightarrow$ train models $\rightarrow$ enhance user capabilities

- Social networking assistance tool: probabilistic model of contacts likely to be relevant (to a part. group) (Amershi et al.)
- IML process to assist auditors for health insurance claims: highlighting of fields of interest and "grouping similar claims in the processing queue to reduce inefficiencies of context switching" (by Ghani and Kumar)

- *AppGrouper* helps clustering topically coherent applications in online app store, where users can edit clusters and encourage creation of more/fewer

"Pairing human judgement with computational intelligence through a carefully designed interface" proves valuable in imporoving the clustering results over alg.-generated clusters f.ex.

System level architecture may not require modifying

4.4 Raw numerical data

Demonstrate data type agnosticism: IML process with numerical data, abstracted from any particular application

- *BrainCel*: adds ML functionality to spreadsheet interface; user selects rows, model makes predictions based on those (Sarkar et al.)
- See paper for more examples

These approaches lose in terms of domain relevance because of their abstraction but gain in flexibility.

4.5 Composition of an IML Interface

Presenting interface elements that are common among the broad range of IML syst.

- Desired attributes of interface:
  - Illustrat. of current state of learned concept
  - Guiding user to provide quality improving input
  - Provide revision mechanisms
- Four key interface elements (paper, Fig. 3.):

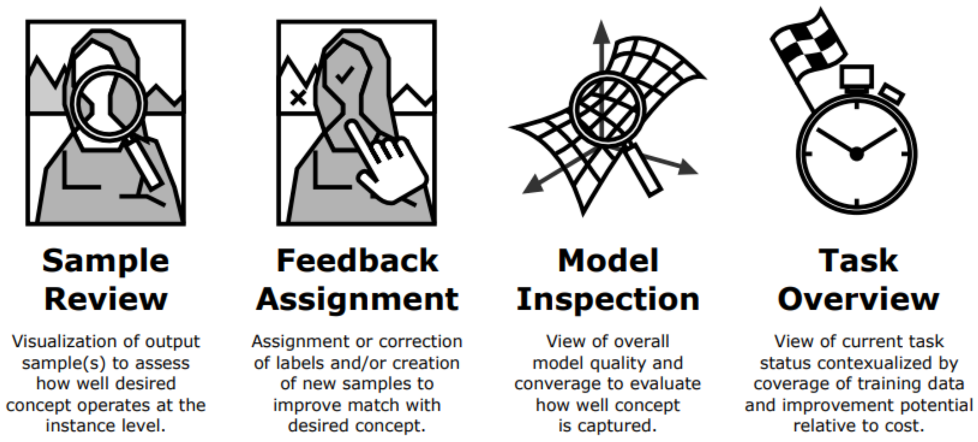

Sample Review
Visualization of output sample(s) to assess how well desired concept operates at the instance level.
Feedback Assignment
Assignment or correction of labels and/or creation of new samples to improve match with desired concept.
Model Inspection
View of overall model quality and converage to evaluate how well concept is captured.
Task Overview
View of current task status contexualized by coverage of training data and improvement potential relative to cost.

Fig. 3. Distinct interface elements in the IML process.

"We do not suggest that these elements are necessarily visually distinct, or indeed necessary in all applications of IML."

"one-time only custom task may have limited or no use for the model inspection or task overview interfaces"

4.5.1 Sample review

- Model outputs are presented for review by user
- Useful: Selection of representative and non-redundant samples for review (users are unwilling to judge large numbers of test cases)
- Examples of displaying output samples:
  - Showing best and worst samples for classification
  - Show regions of the image belonging to classification
- Challenge: Specific output samples doesn't always reflect the generalised model performance

Interesting: "[...] people want to understand why specific instances fail. Understanding why the model fails for a given sample may help the user determine the most appropriate feedback strategy"

$\Rightarrow$ Avoid overly abstract/condensed representat. of output samples

4.5.2 Feedback assignment

- User assigns labels, selects features, generates new samples
- Perhaps most crucial interface
- Challenge: Concept that users are trying to train may shift over time $\rightarrow$ efforts to improve guidance for being requested and providing input

4.5.3 Model inspection

Summative view of model quality

- ORES metrics: ... ? "The assessment of model quality is not necessarily limited to prediction accuracy, and may also incorporate other metrics such as coverage and confidence."
- Three main causes of model errors:
  1. Mislabelled data
  2. Feature deficiencies: not enough samples of given feature for model to make a distinction
  3. Insufficient data
- Model inspection interface: support user in identifying such errors
- Also often used to evaluate model performance: ROC and cost curves, but also targeted testing $\leftarrow$ difficult for end-users
- Visual inspection of model performance may be complicated with data formats $\neq$ images; one solution might be *ModelTracker*, a visual representation of model performance for use in debugging by Amershi et al.

Note: "framing task completion criteria purely in terms of model quality is undesirable. Focusing an end user on model prediction accuracy alone is likely to promote overfitting. For this reason, we make a distinction between the model inspection and task overview interfaces."

4.5.4 Task overview

- Only sometimes included
- Provide visibility of global objectives
- Contextualise objectives with information, such as availability of training data and target application
- Guide user recognizing the most efficient way of improving the model (improving classification accuracy is not always the answer!)

4.6 A Generalised Workflow for a Comprehensive IML Process

IML worflow as behavioural breakdown into six user activities:

1. Feature selection
2. Model selection
3. Model steering ← **core activity, majority of time spent here**
4. Quality assessment
5. Termination assessment
6. Transfer

"to be interpreted in combination with the structural breakdown of the IML interface" (Sample Review, Feedback Assignment, ...)

4.6.1 Feature selection

- This step may be unnecessary in many applications, but ...
- ... users selecting features delivers efficiency and quality gains.
- Also: users enjoy "identifying useful features for incorporation in the training process".

"Designing the IML system and interface to accommodate the feature selection activity may thus be desirable, though not always practical" (certain data types are not suited for it).

4.6.2 Model selection

- Cases where data and feature structures are not known a priori: allow user to make selection of model (or perform comparative evaluation **and not** the author)
- Another argument for it: ideal IML system is agnostic to underlying model and algorithm
- User experimenting with multiple different ML techniques within IML app. ⇒ learning process without an expert

4.6.3 Model steering

- Interaction involved:
  - Correcting samples
  - Assigning labels to new samples
  - Generating completely new samples
- "The emphasis, however, is generally on providing <u>just enough</u> feedback per iteration to push the model in the right direction."
- Quality of this dialogue is directly influenced by quality of interface design
- Challenge: users seem to not like undergoing multiple iterations, but instead spend more time labelling samples → need for high-level guidance (e.g. follow iterative approach!)
- Even more so since users suffer from inconsistency (⇒ *Structured Labeling* by Kulesza et al.)

4.6.4 Quality assessment

- Evaluate the trained system performance
- After some training, user will periodically check model quality and compare against "some desired or pre-established threshold".
- Challenges in this section: Support users in selecting appr. strategy and avoiding overfitting

4.6.5 Termination assessment

- Potentially some overlap with **Quality assessment**
- User should decide when to end IML process, ...
- ... but alternative conditions for ending it could be:
  - further labelling will deliver negligible gain
  - there is already sufficient confidence in the particular classifications of value
  - increasing risk of overfitting

  User could base his decision on contextualisations like those

4.6.6 Transfer

- **Transfer** = Deploying trained model in target domain
- Tools are often applied in unpredictable (at design time) ways
- Some applications do not require transfer activity
- Still useful to consider end use of trained models and its impact on the interface design

4.7 Emergent Solution Principles

5 Possible goal of the thesis

Questions

- Q: Should I ask Aaron how he would like us to work together?

  A:

- Q: In what situations exactly do we want to optimize the threshold in the context of user centered threshold optimization?

  A: