Machine Learning For Design

Lecture 7 - Designing And Develop Machine Learning Models / Part 1

Alessandro Bozzon
Yen-Chia Hsu
16/03/2022

mlfd-io@tudelft.nl
www.ml4design.com

Admin

Few remarks

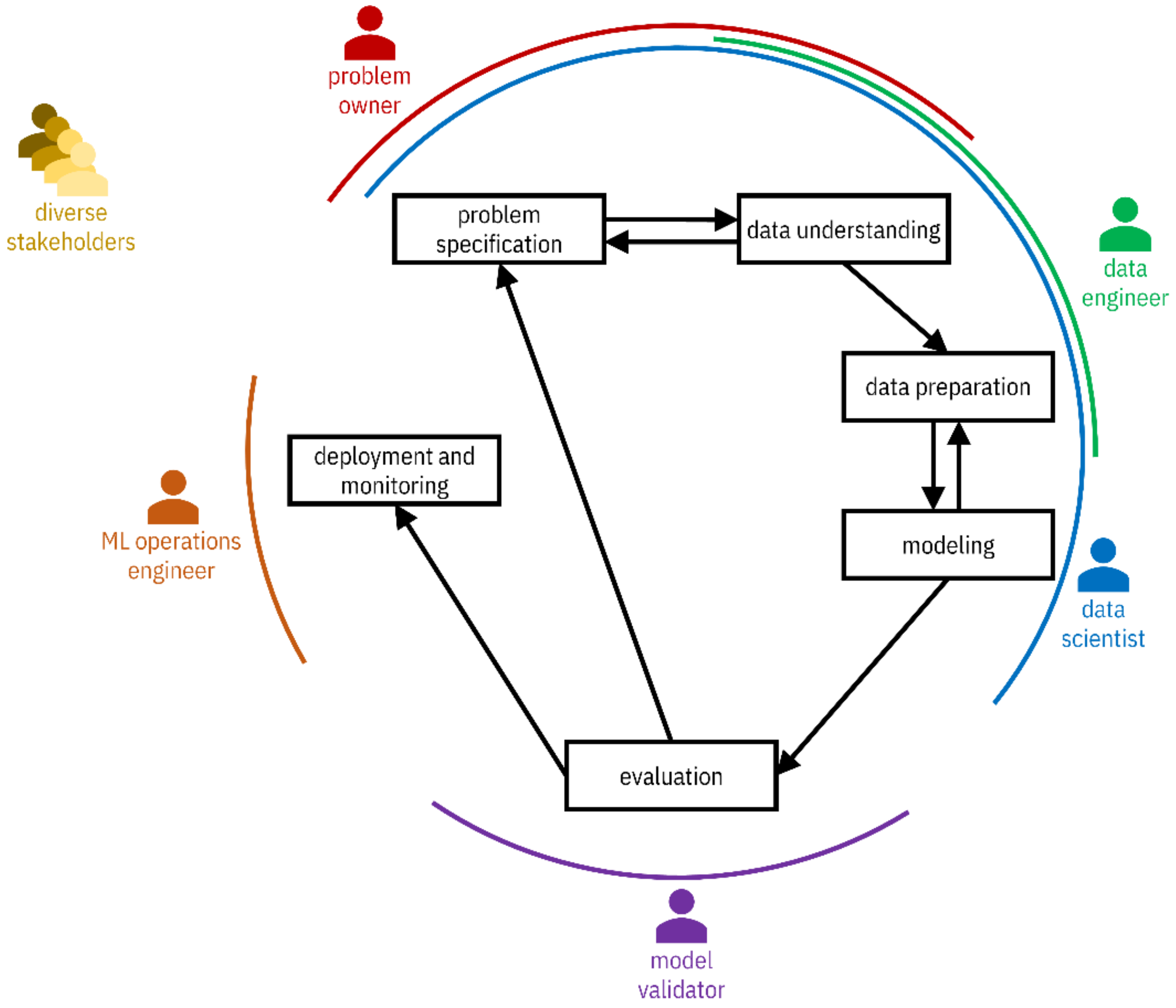
- All assignments 2 are in
- We are looking forward to reading them!

- Thank you for your feedback on Module 2!

Previously, on ML4D….

Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology

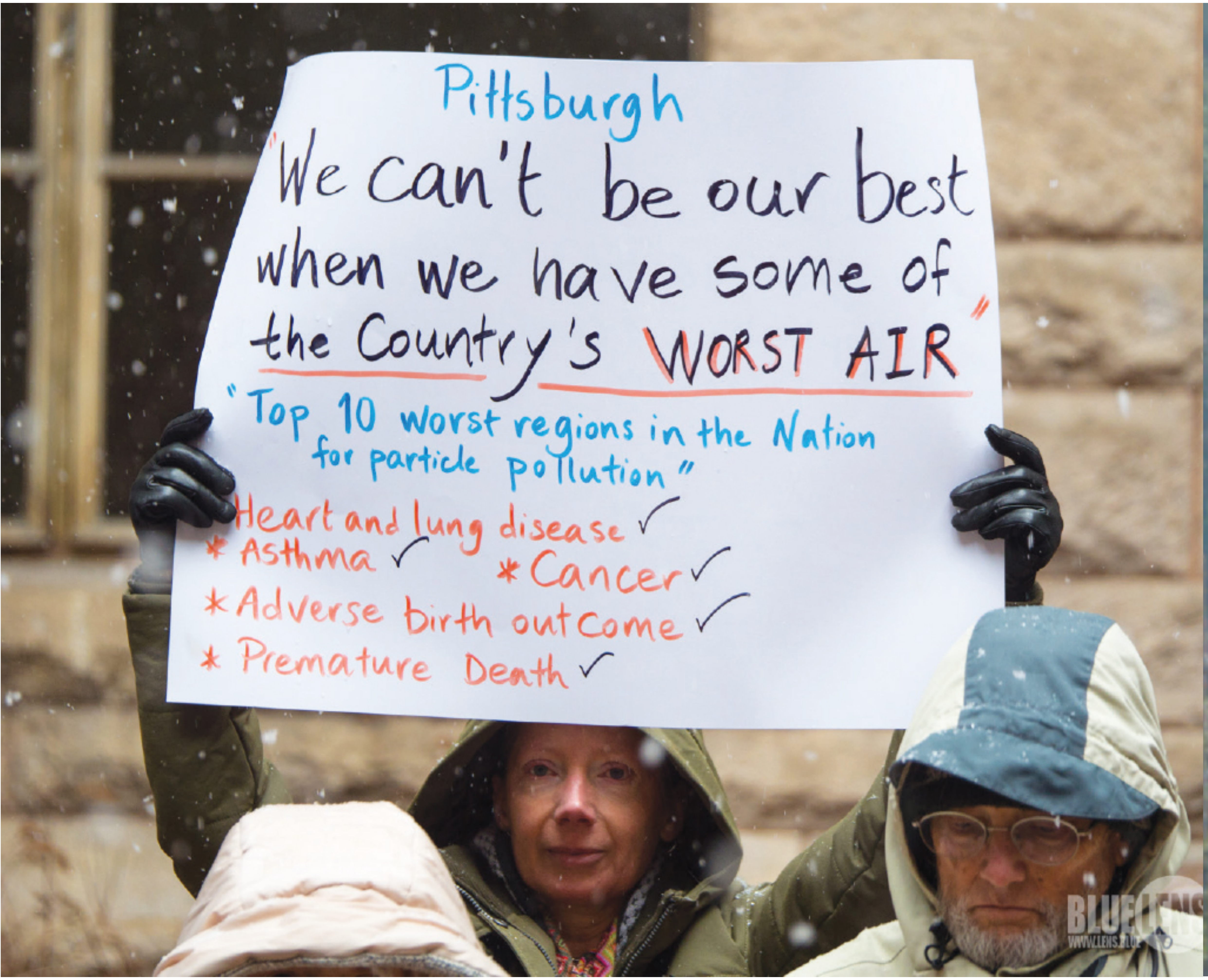
Lecture 2



https://www.the-modeling-agency.com/crisp-dm.pdf

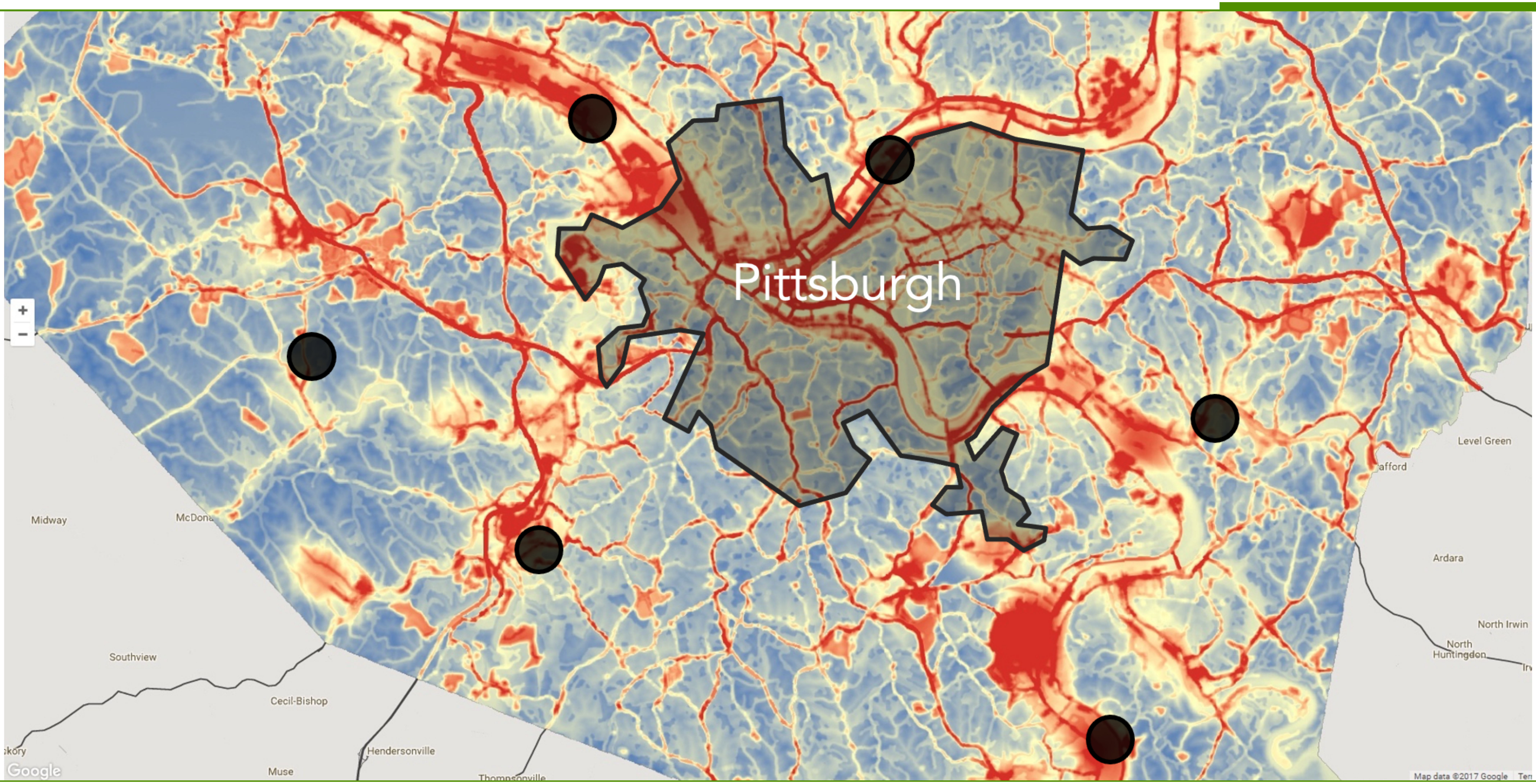
Let’s go to Pittsburgh

According to the American Lung Association, Pittsburgh is one of the ten most polluted cities (measured by particulate matter) in the United States. Local residents have been fighting against air pollution for decades.



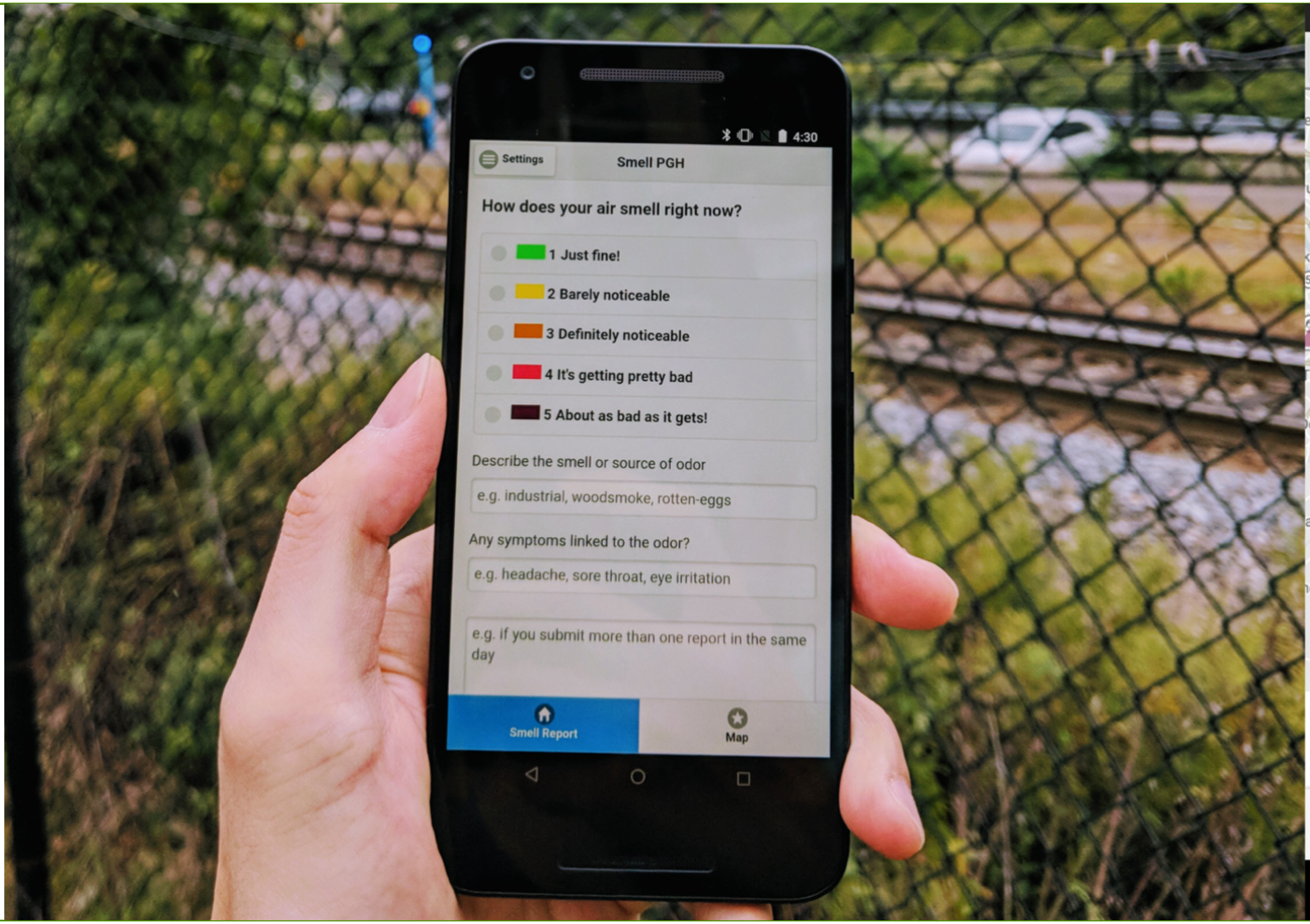
Link to the American Lung Association — https://www.lung.org/research/sota/key-findings/year-round-particle-pollution

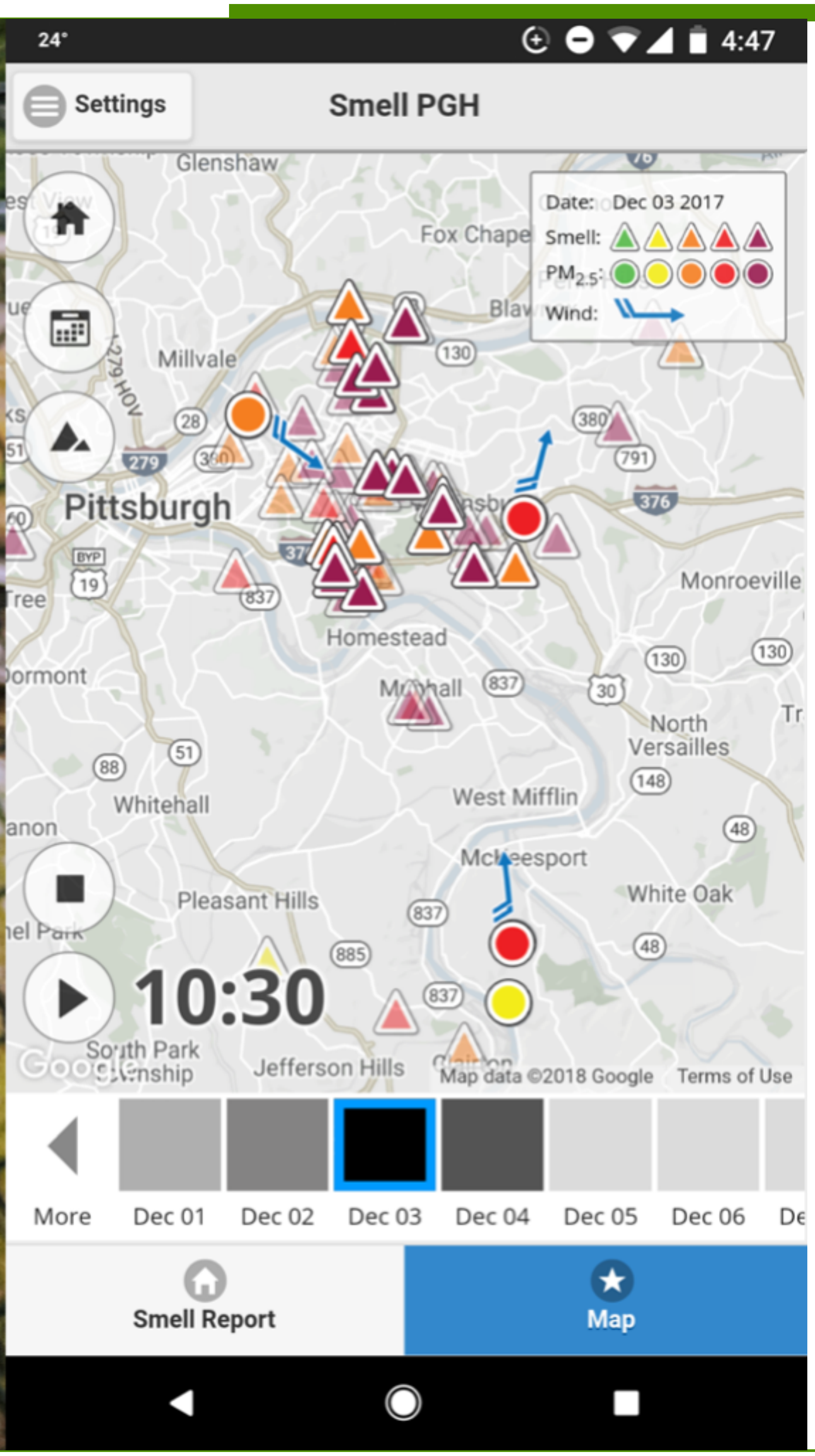
Local people have identified smell as an indicator of air pollution. But, how can we effectively collect the smell experiences on a city-wide scale with more than 300,000 residents over many years?



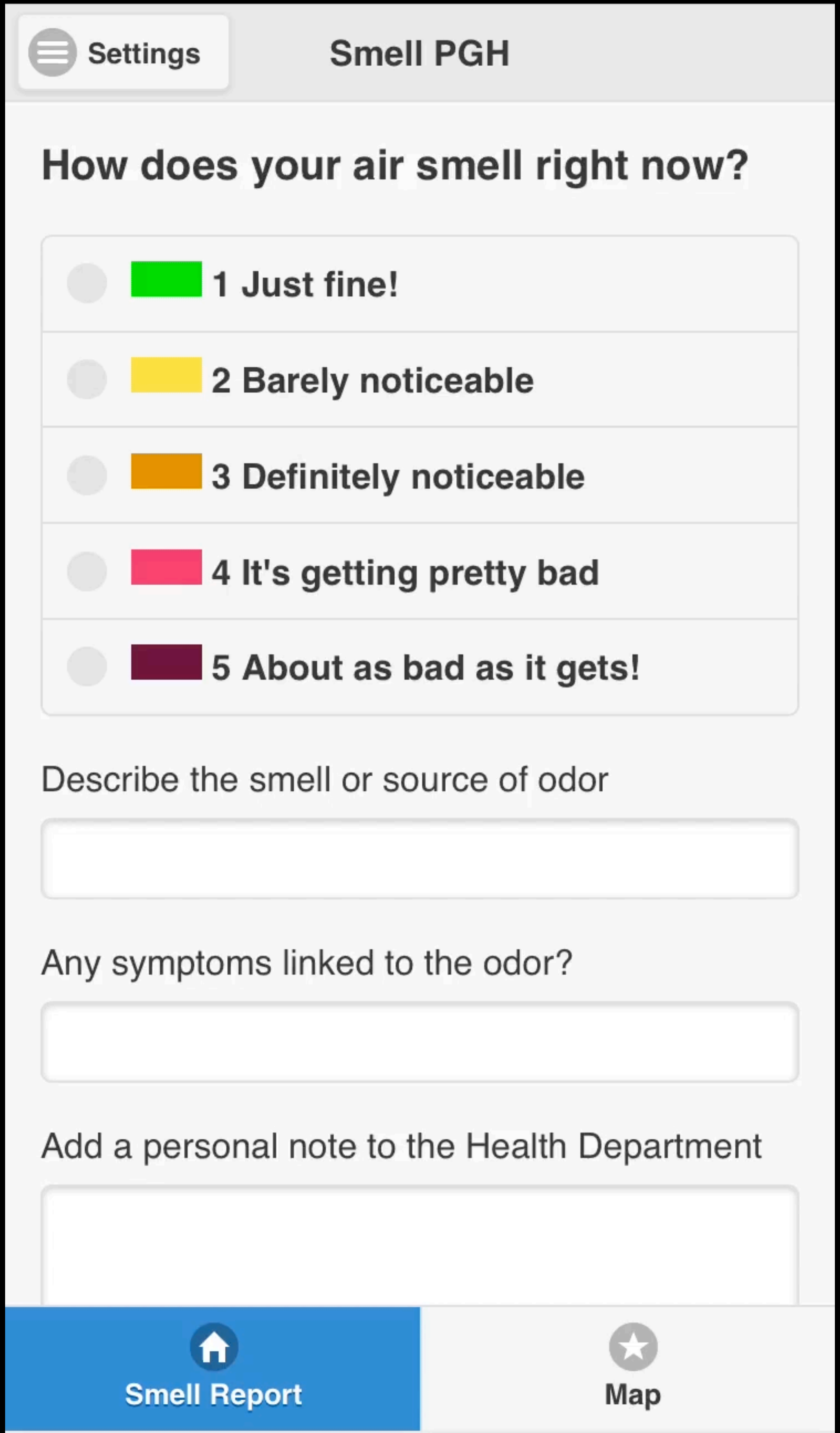
Link to the Pittsburgh pollution map — https://breatheproject.org/pollution-map/

Smell Pittsburgh is a mobile application that enables local communities to contribute odor reports in real-time (with accurate time and location information) and visualize air pollution collaboratively.





Link to the Smell Pittsburgh application — https://smellpgh.org


Settings
Smell PGH
How does your air smell right now?
1 Just fine!
2 Barely noticeable
3 Definitely noticeable
4 It's getting pretty bad
5 About as bad as it gets!
Describe the smell or source of odor
Any symptoms linked to the odor?
Add a personal note to the Health Department
Smell Report
Map

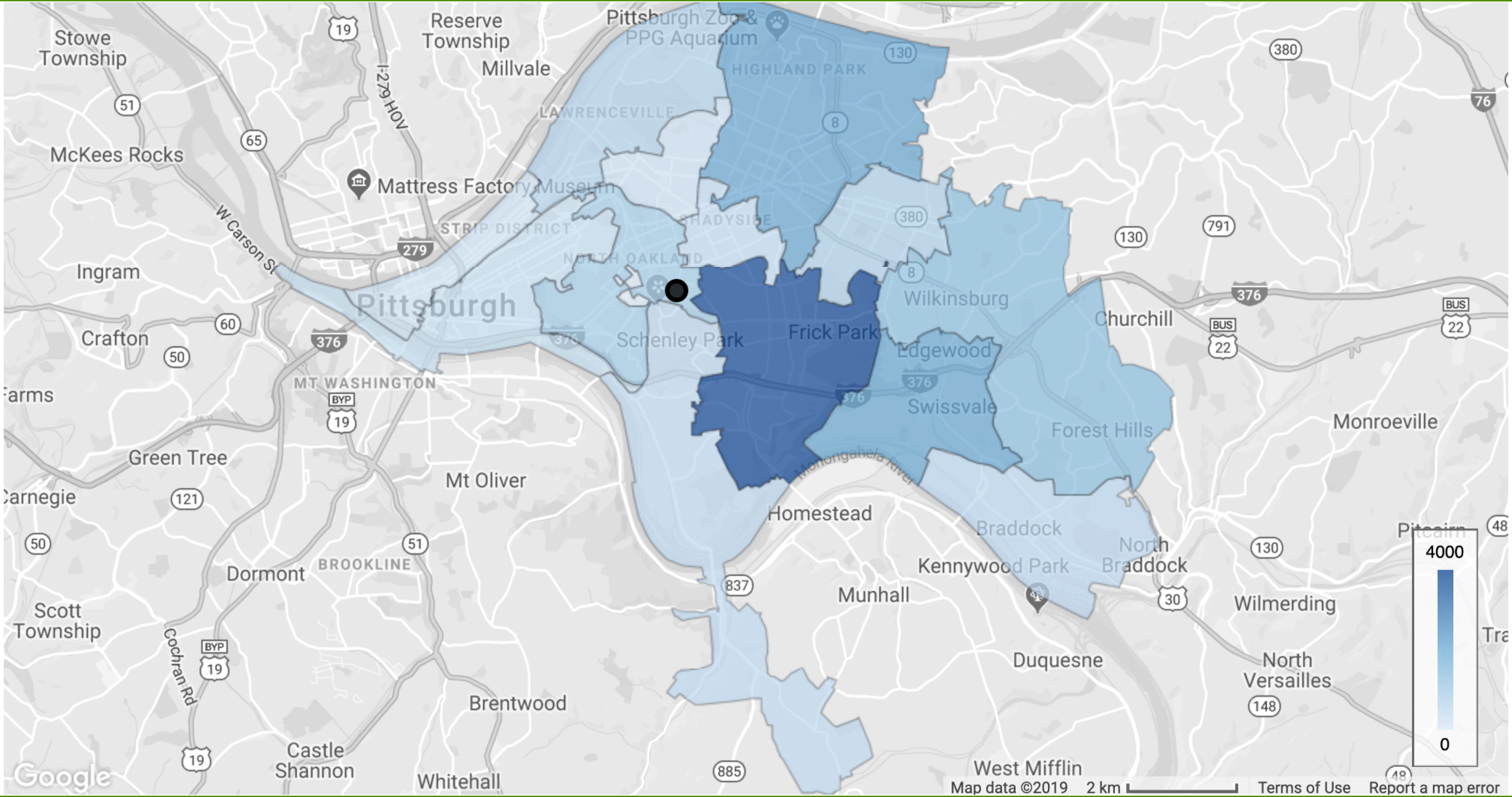
Smell Pittsburgh predicts upcoming smell events (based on the existing data at a certain time point) and sends push notifications to inform users while encouraging engagement in submitting odor data.

SMELL PGH

Smell Event Alert

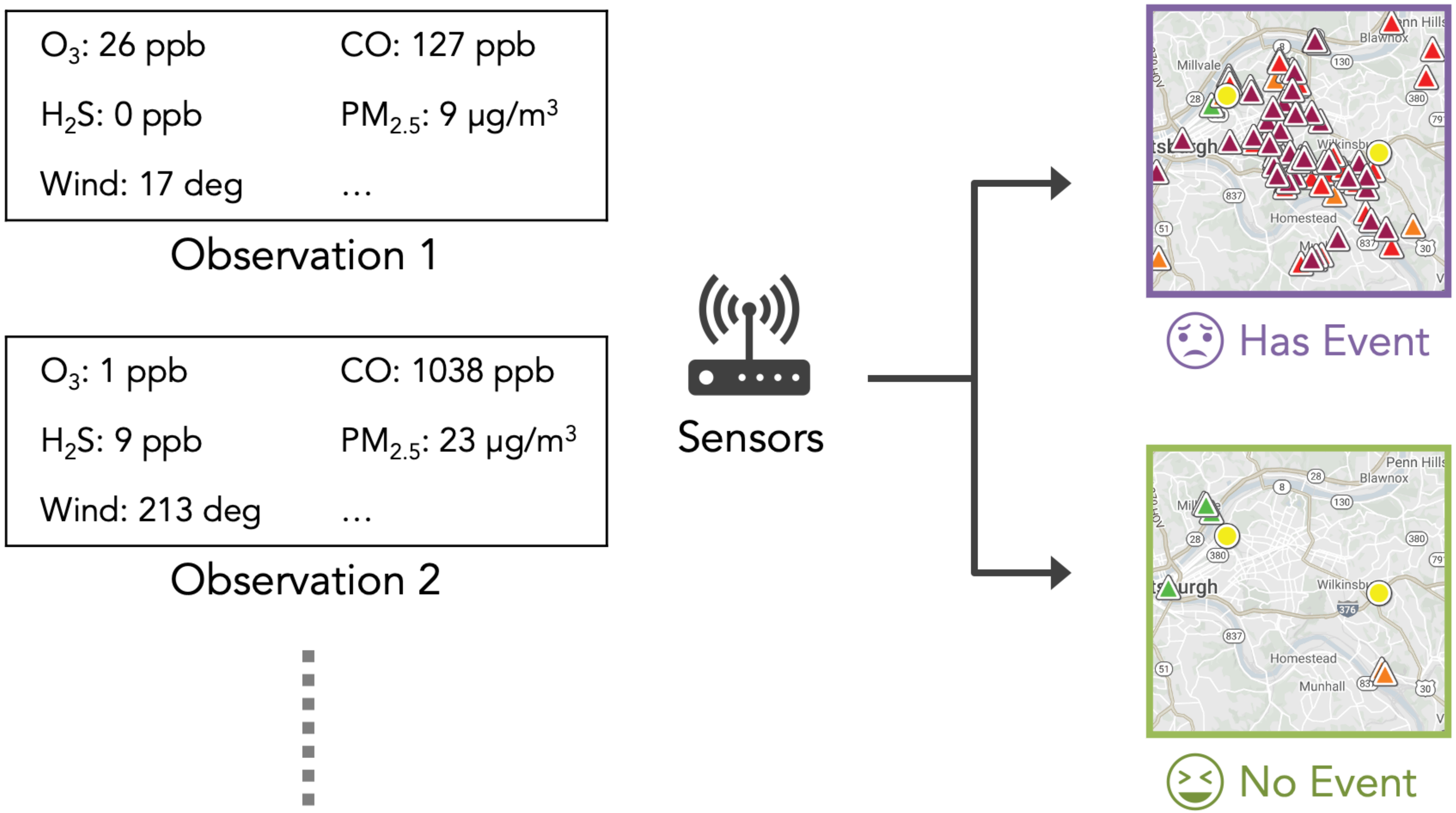
Local weather and pollution data indicates there may be a Pittsburgh smell event in the next few hours. Keep a nose out and report smells you notice!

A geographic region in Pittsburgh is manually selected when predicting the smell events. The black dot in the figure represents the location of Carnegie Mellon University.



Number of smell reports aggregated by zip codes in the dataset.

To predict the presence of bad odor within the next few hours, we need to estimate a function that can map sensor measurements to smell events as accurately as possible.



One can technically use if-else rules to predict smell events. But such an approach can be laborious. Can we do better than manually specifying these if-else rules while minimizing human efforts?

O_3: 26 ppb	CO: 127 ppb
H_2S: 0 ppb	$PM_{2.5}$: 9 μg/m^3
Wind: 17 deg	…

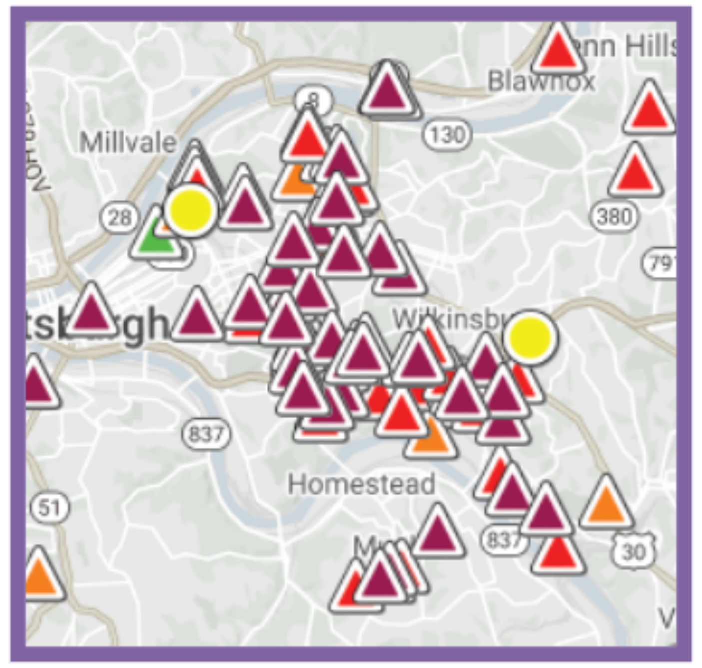
Observation 1

O_3: 1 ppb	CO: 1038 ppb
H_2S: 9 ppb	$PM_{2.5}$: 23 μg/m^3
Wind: 213 deg	…

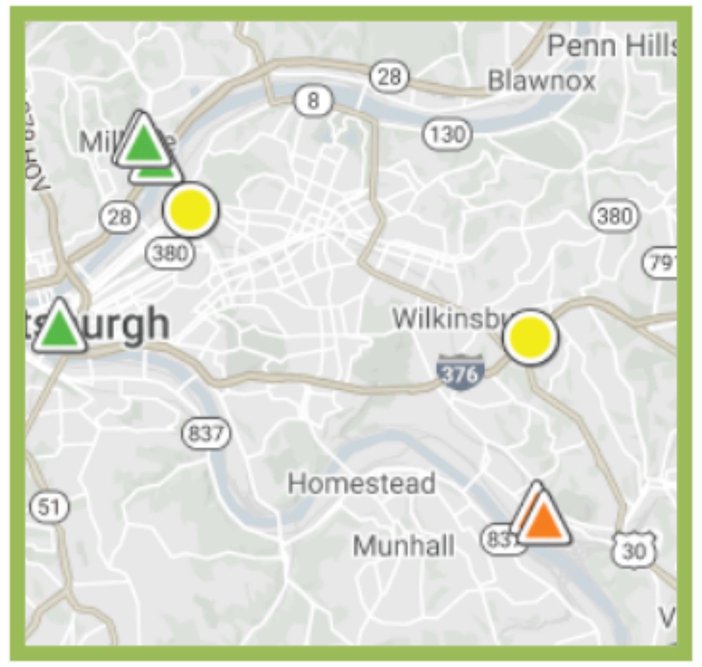
Observation 2

if H_2S > ?
and CO > ?
and $PM_{2.5}$ > ?
and …
then has event

else no event

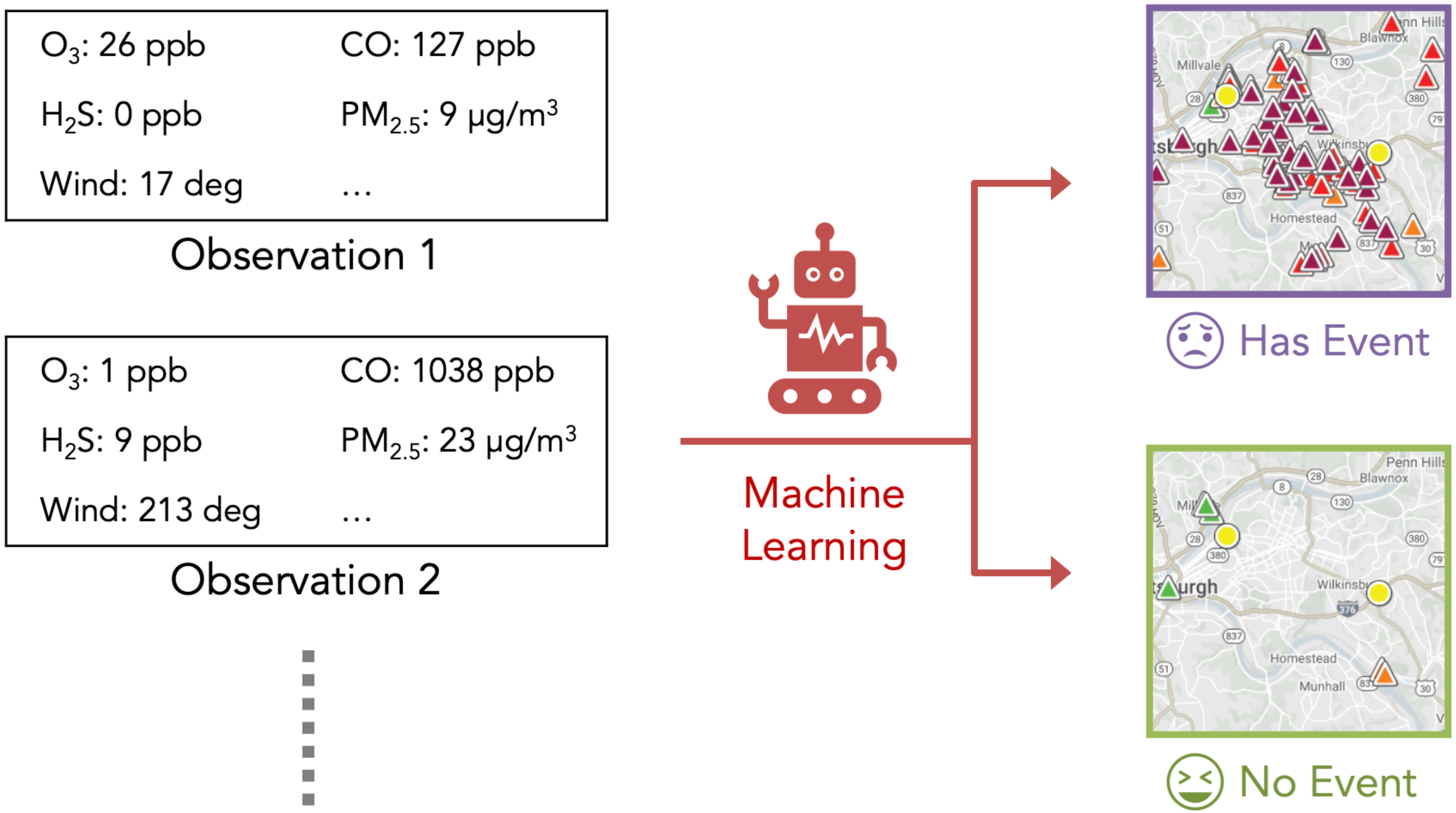


Has Event



No Event

It turns out that we can use the Smell Pittsburgh dataset to estimate a function (i.e., train a machine learning model) that can predict smell events from sensor measurements.



Researchers collected the Smell Pittsburgh dataset, including all the smell reports and sensor measurements (from air quality and weather monitoring stations) from October 31 in 2016 to September 30 in 2018.

- Samples of Citizen-Contributed Smell Reports

EpochTime	feelings_symptoms	smell_description	smell_value	zipcode
…	…	…	…	…
1478353854	Headache, sinus, seeping into house even though it is as shut and sealed as possible. Air purifiers are unable to handle it thoroughly.	Industrial, acrid, strong	4	15206
1478354971		Industrial	4	15218
…	…	…	…	…

- Samples of Air Quality Sensor Measurements

EpochTime	3.feed_28.H2S_PPM	3.feed_28.SO2_PPM	3.feed_28.SIGTHETA_DEG	3.feed_28.SONICWD_DEG	3.feed_28.SONICWS_MPH
…	…	…	…	…	…
1478046600	0,019	0,020	14,0	215,0	3,2
1478050200	0,130	0,033	13,4	199,0	3,4
…	…	…	…	…	…

Link to the Smell Pittsburgh data — https://github.com/CMU-CREATE-Lab/smell-pittsburgh-prediction


ONE DOES NOT SIMPLY
USE RAW DATA IN MACHINE LEARNING
imgflip.com

We need to quantitatively define a smell event (i.e., the presence of bad odor): whether the sum of smell values within a specific time range is larger than a particular threshold.

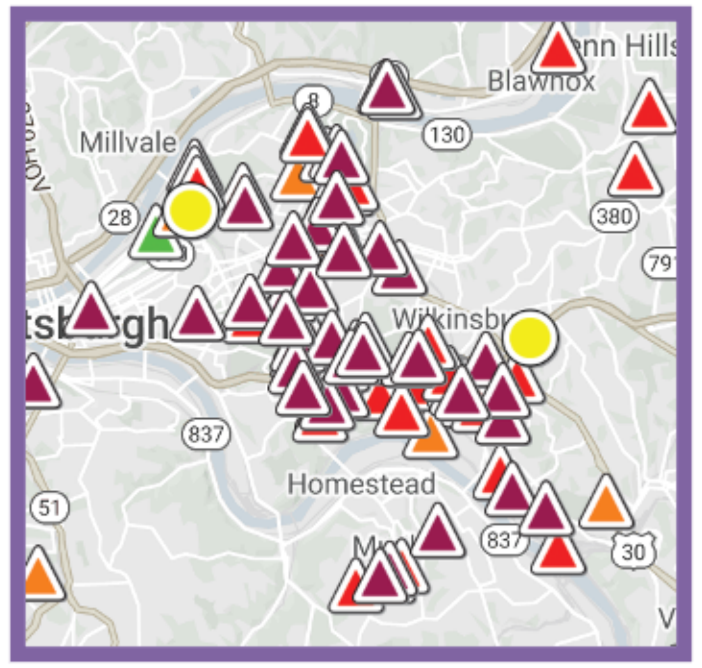
- Samples of Citizen-Contributed Smell Reports

EpochTime	smell_value	zipcode
...	...	...
1478353854	4	15206
1478354971	4	15218
1478359473	4	15218
1478371179	3	15207
1478393585	3	15217
1478399011	4	15217
1478432399	4	15218
1478432502	2	15206
1478434105	4	15217
1478435133	4	15206
1478435313	4	15206
1478435748	3	15206
1478435801	5	15218
...	...	...

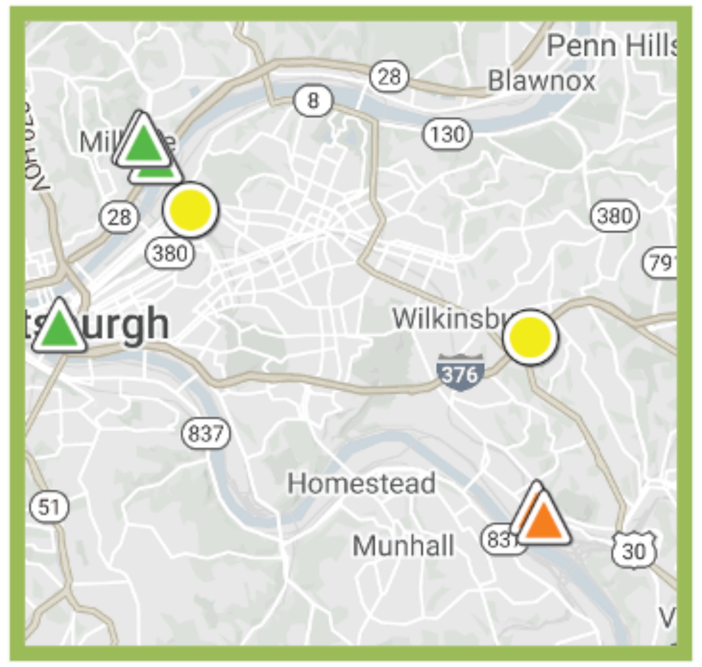
if the sum of smell values
within H hours > V
(need to define H and V)

then has event

else no event

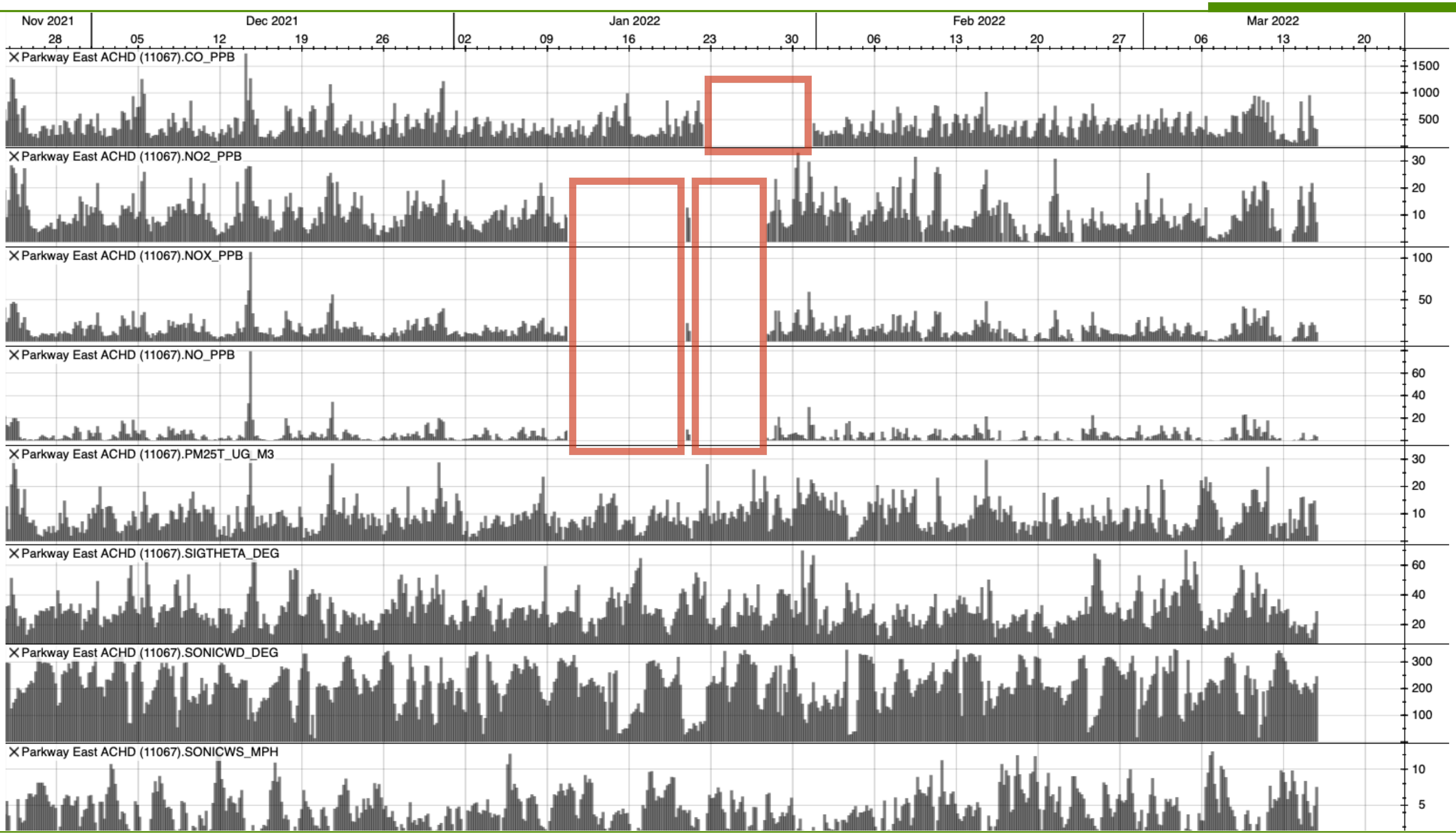


Has Event

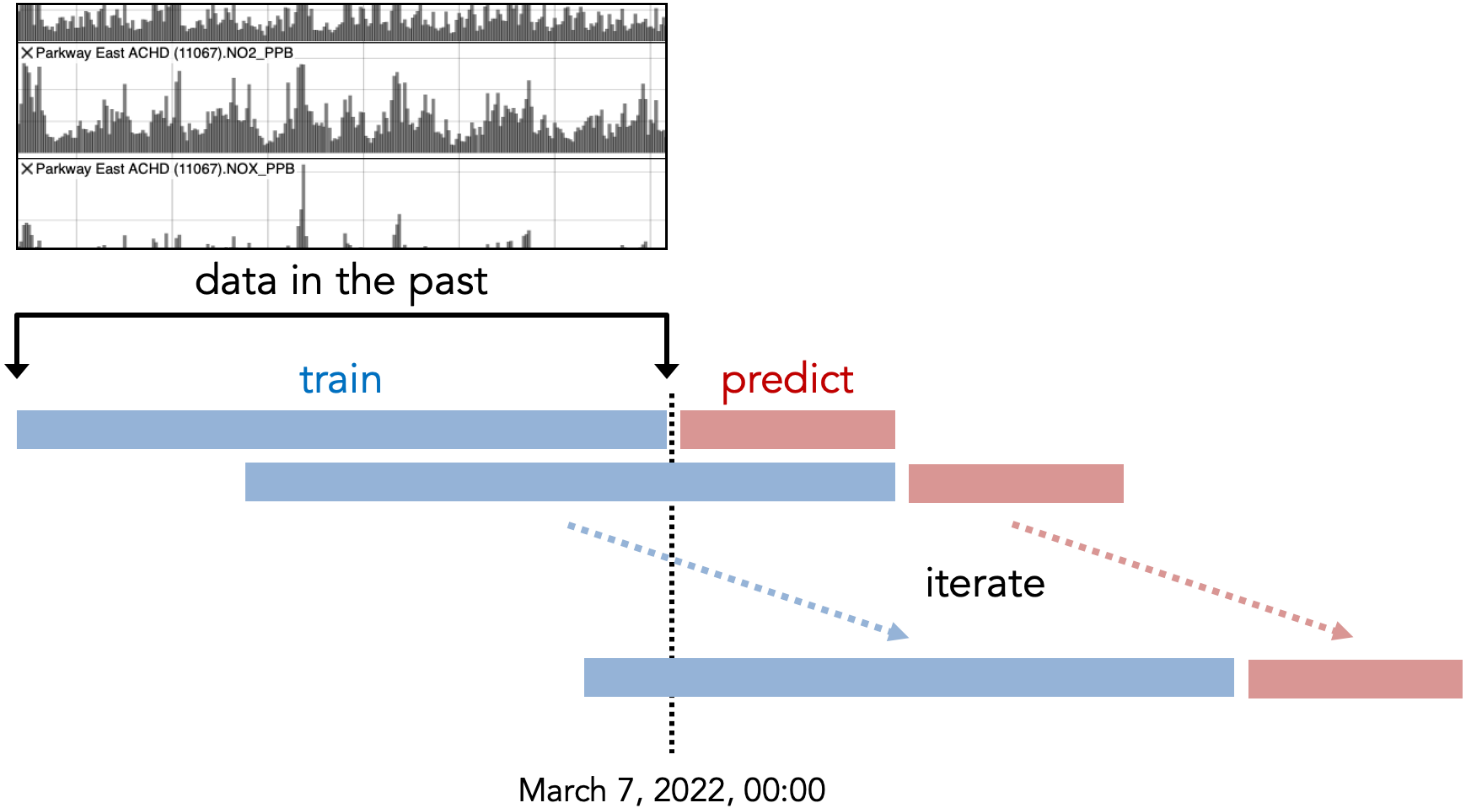


No Event

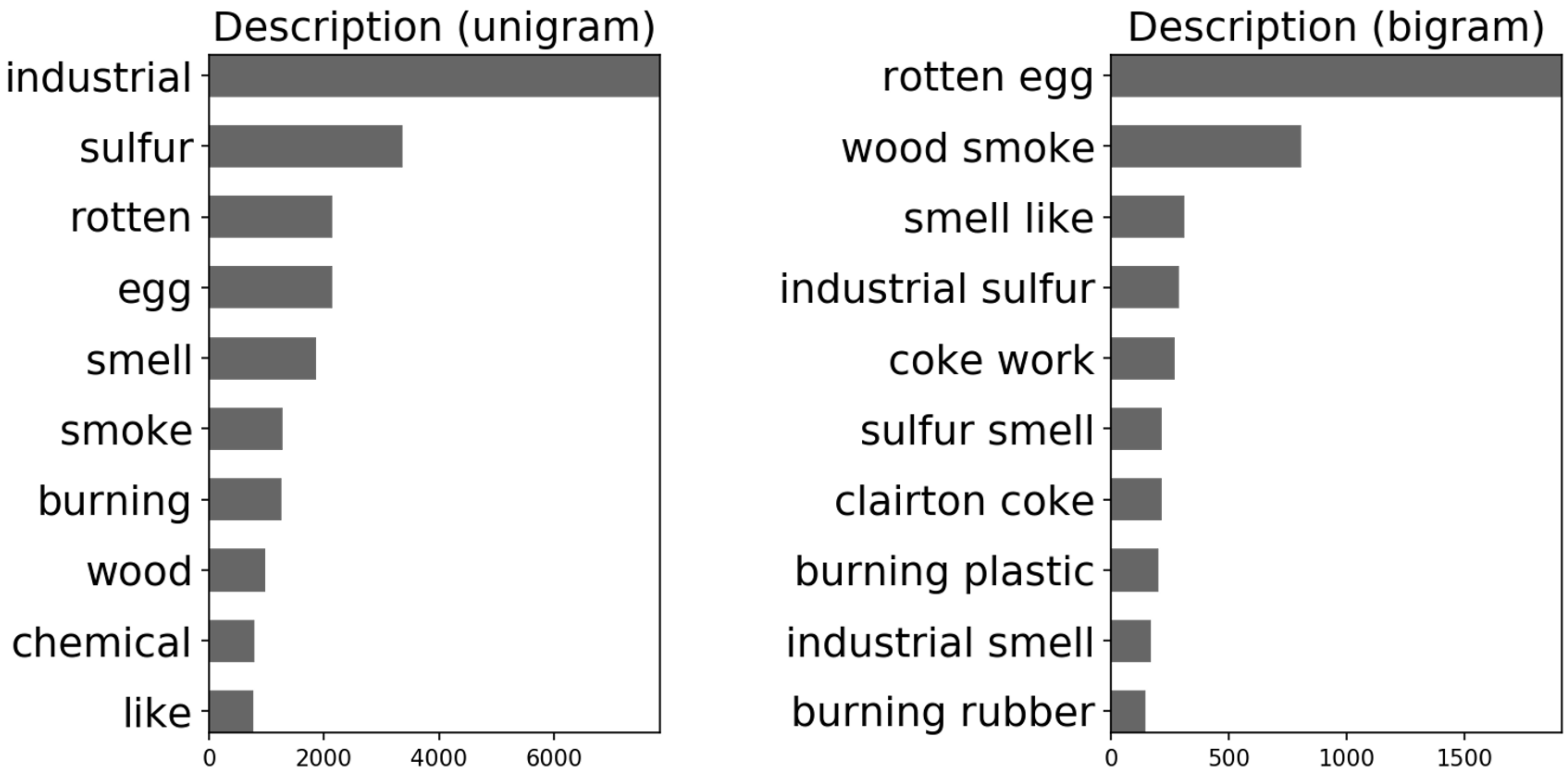
We need to treat missing data. The sensor measurements can be missing during some time periods since some air quality or weather monitoring stations may be down for maintenance.



The dataset contains time-series data, which means each data point has a timestamp, and we can only use data in the past (i.e., data that exists for a specific time point) to train the model to predict the future.



How do we know which variables from which monitoring stations are effective in predicting the presence of bad odor? We can explore the data to get insights or rely on local knowledge of pollution sources.



We also need to extract and decide the features that we want to use when training the machine learning model. Such features can help us identify air pollution patterns in the Pittsburgh region.

Node 1
# of positive : negative samples
= 316 : 16,378

Is Lawrenceville north-south wind direction (current hour) times Liberty H_2S (previous 1 hour) > 2.19?

- No → (...)
- Yes →

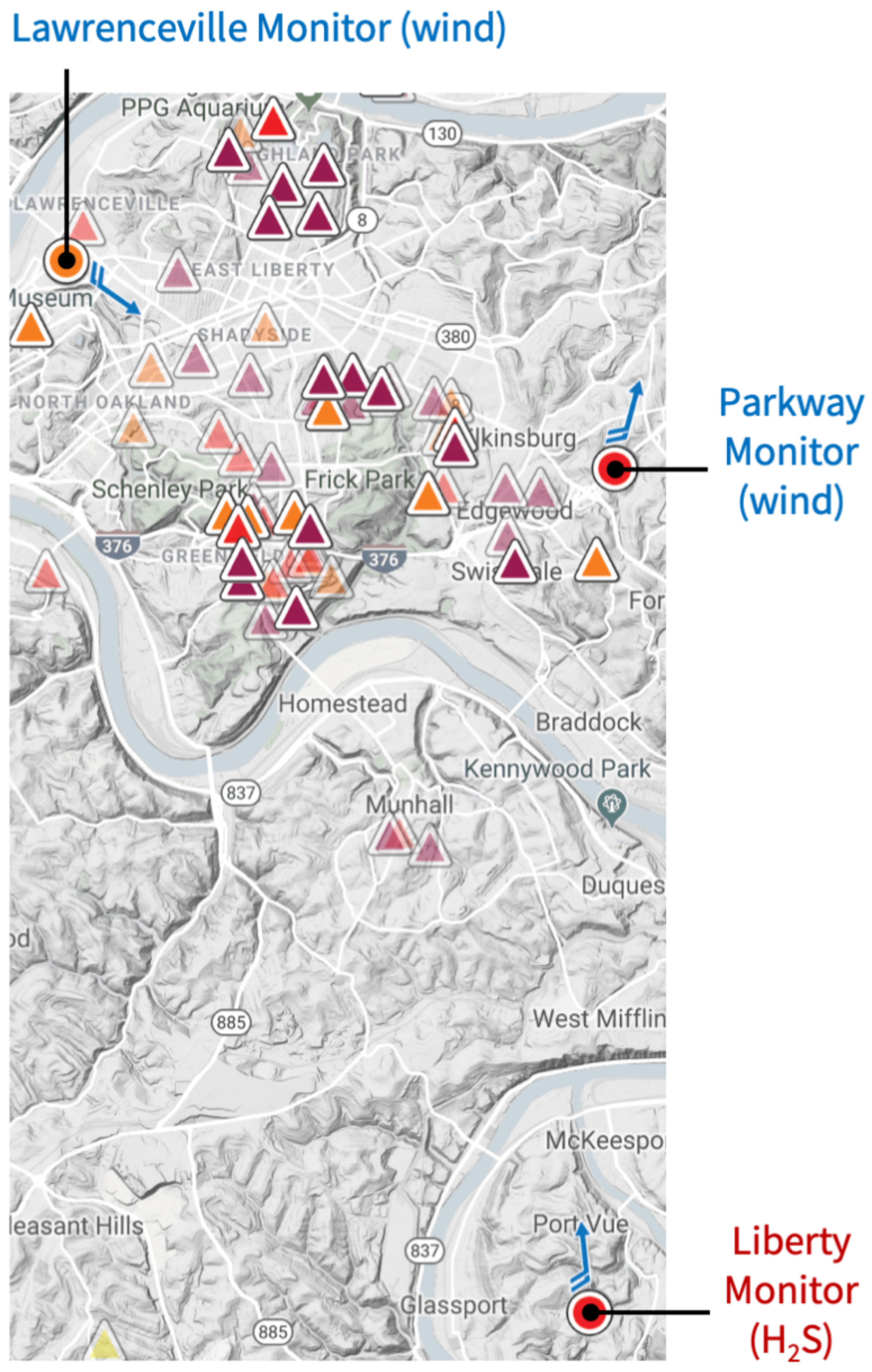
Node 2
# of positive : negative samples
= 261 : 363

Is Parkway east-west wind direction (current hour) times Liberty H_2S (current hour) > 2.86?

- No → (...)
- Yes →

Node 3
# of positive : negative samples
= 154 : 74

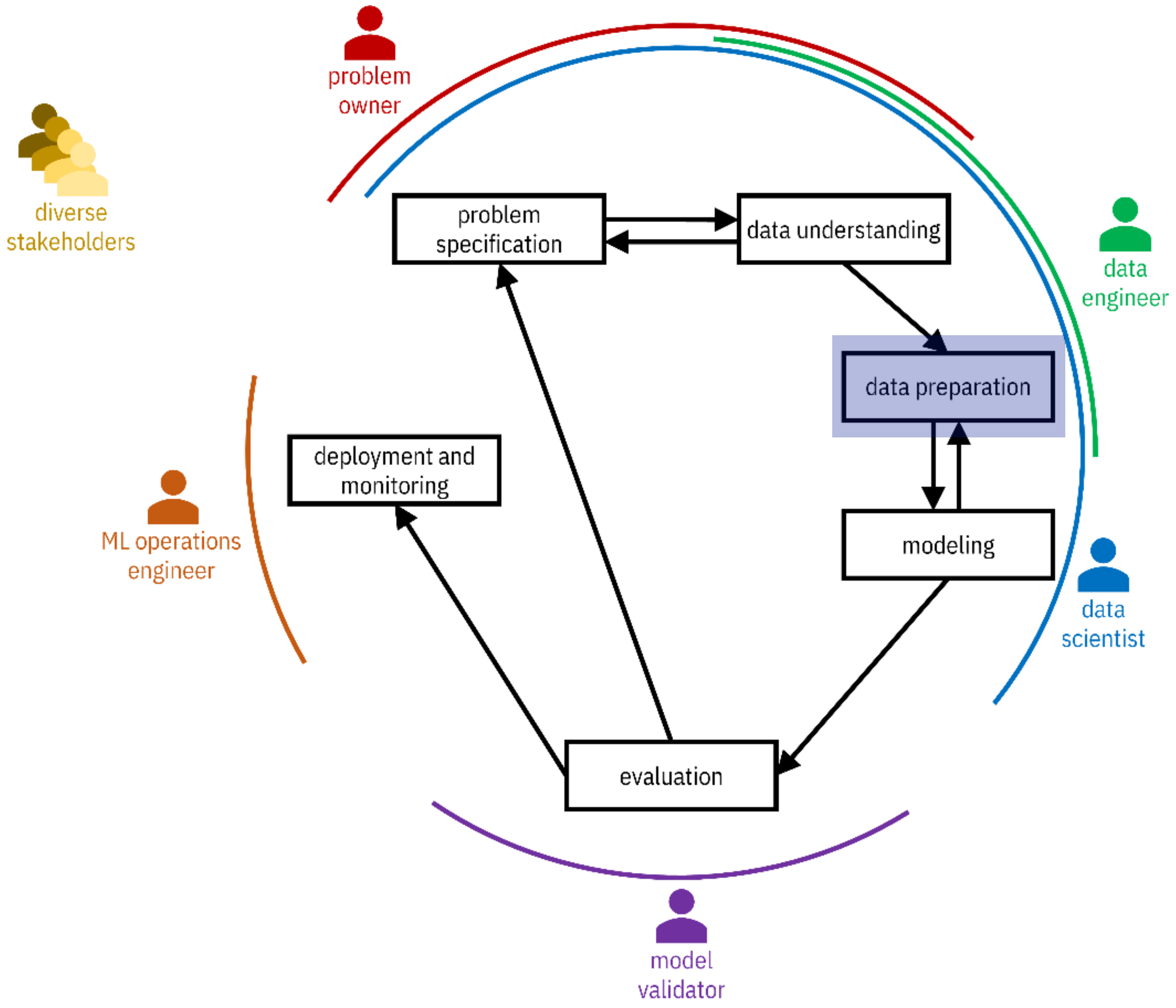
(...)



Data Preparation

Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology

Lecture 2



https://www.the-modeling-agency.com/crisp-dm.pdf

Types of Feature / Label Values

Lecture 2

Categorical

- Named data
- Can take numerical values, but no mathematical meaning

Nominal

- No order
- No direction

  - **Marital status**
  - **Gender**
  - **Ethnicity**

Ordinal

- Order
- Direction

  - **Letter grades (A, B, C, D)**
  - **Socio-economic status (poor, rich)**
  - **Ratings (dislike, neutral, like)**

Numerical

- Measurements
- Take numerical values
  - Discrete or continous

Interval

- Difference between measurements
- No true zero or fixed beginning

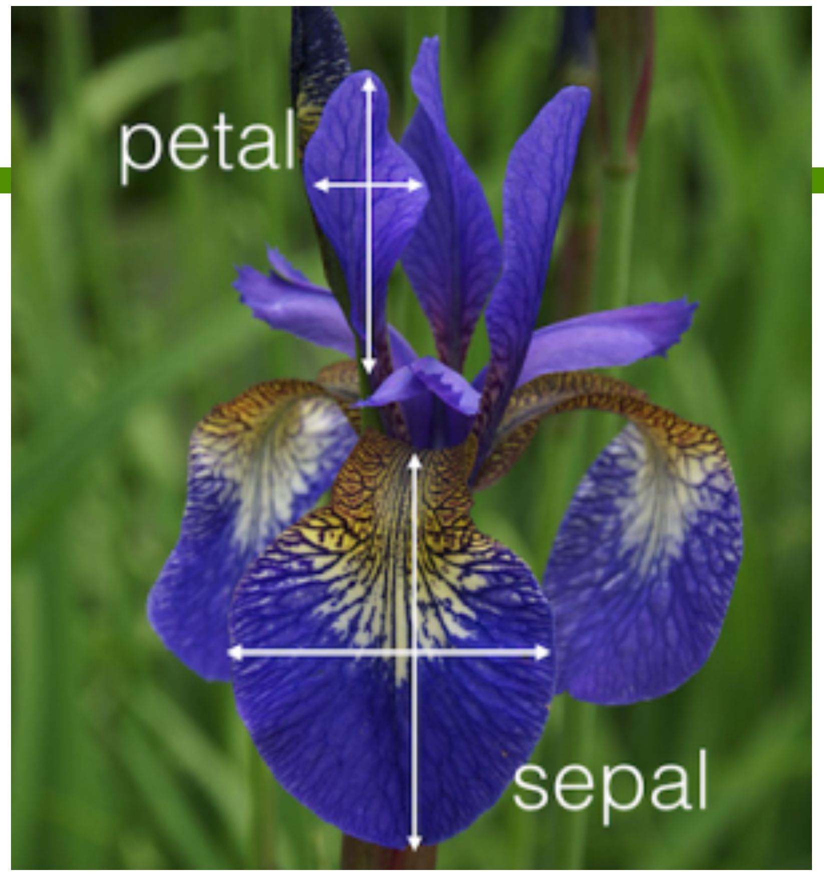
  - **Temperature (C or F)**
  - **IQ**
  - **Time, Dates**

Ratio

- Difference between measurements
- True zero exists

  - **Temperature (K)**
  - **Age**
  - **Height**
  - **Weight**

Ideal Data



Numerical Feature | Numerical Feature | Numerical Feature | Numerical Feature | Label

sepal_lenght	sepal_width	petal_lenght	petal_width	Class
5.0	3.3	1.4	0.2	Iris-setosa
7.0	3.2	4.7	1.4	Iris-versicolor
5.7	2.8	4.1	1.3	Iris-versicolor
6.3	3.3	6.0	2.5	Iris-virginica

Record / Sample / Data Item

Label Value

Feature Value

Dataset Size

Dataset Dimensionality

https://archive.ics.uci.edu/ml/datasets/iris

Mixed Feature Types

- Data is rarely "clean"
  - Approximately 50-80% of the time is spent on **data wrangling** - could be an under-estimate
- Having good data with the correct features is absolutely critical
- 3 issues to deal with:
  - *Encoding features* as numerical values
  - *Transforming features* to make ML algorithms work better
  - Dealing with *missing feature values*

MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	...	MoSold	YrSold	SaleType	SaleCondition	SalePrice
20	RL	80.0	10400	Pave	NaN	Reg	...	5	2008	WD	Normal	174000
180	RM	35.0	3675	Pave	NaN	Reg	...	5	2006	WD	Normal	145000
60	FV	72.0	8640	Pave	NaN	Reg	...	6	2010	Con	Normal	215200
20	RL	84.0	11670	Pave	NaN	IR1	...	3	2007	WD	Normal	320000
60	RL	43.0	10667	Pave	NaN	IR2	...	4	2009	ConLw	Normal	212000
80	RL	82.0	9020	Pave	NaN	Reg	...	6	2008	WD	Normal	168500
60	RL	70.0	11218	Pave	NaN	Reg	...	5	2010	WD	Normal	189000
80	RL	85.0	13825	Pave	NaN	Reg	...	12	2008	WD	Normal	140000
60	RL	NaN	13031	Pave	NaN	IR2	...	7	2006	WD	Normal	187500

Categorical features

Ordinal features

Numeric features

Looks numeric, but is actually categorical

Easy case: features are already numerical (or Boolean)

- Each feature is assigned its own value in the feature space

IsAdult	Age
FALSE	17
TRUE	21
TRUE	34
FALSE	9

IsAdult	Age
0	17
1	21
1	34
0	9

One-hot encoding of categorical features

- Why not encode each value as an integer?
  - A naive integer encoding would create an ordering of the feature values that do not exist in the original data
  - You can try direct integer encoding if a feature does have a natural ordering (ORDINAL e.g. ECTS grades A–F)

- Each value of a categorical feature gets its own column

Status	Gender
Single	M
Married	F
Single	O
Single	M

Status Single	Status Married	Gender M	Gender F	Gender O
1	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	0	1	0	0

Encoding Ordinal Features

- Convert to a number, preserving the order
    - [low,medium,high] —> [1,2,3]
- *Encoding may not capture relative differences*

Health Status	Blood Pressure
Good	Very good
Very Good	Excellent
Normal	Good
Bad	Normal

Health Status	Blood Pressure
3	4
4	5
2	3
1	1

Data Issues

- Incorrect feature values
  - Typos
    - e.g., color = {“blue”, “green”, “gren”, “red”}
  - Garbage
    - e.g., color = “w?r--śiį”
  - Inconsistent spelling (e.g., “color”, “colour”) or capitalization
  - Inconsistent abbreviations (e.g., “Oak St.”, “Oak Street”)

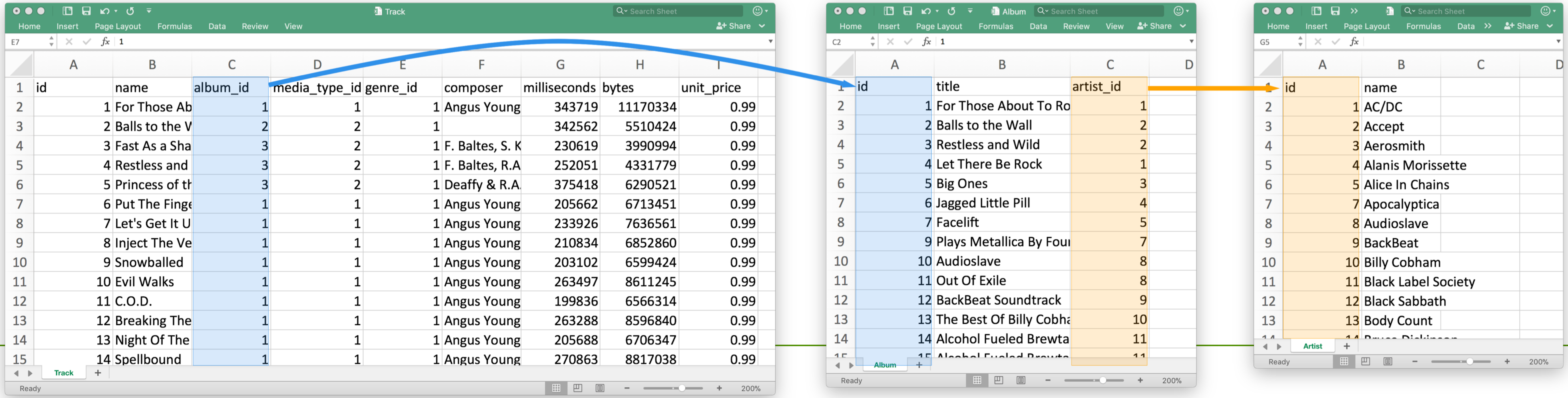
- Missing labels
  - Delete instances if only a few are missing labels
  - Use semi-supervised learning techniques
  - Predict the missing labels via self-supervision

Merging Data

- Data may be split across different files
  - Requires doing a join based on a key to combine data into one table

- Problems During Merge
  - Inconsistent data
    - Same instance key with conflicting labels
    - Data duplication
  - The merged table may be too large for memory
  - Encoding issues
    - Inconsistent data formats or terminology
    - Key aspects mentioned in cell comments or auxiliary files

tracks albums artists



id	name	album_id	media_type_id	genre_id	composer	milliseconds	bytes	unit_price
1	For Those Ab	1	1	1	Angus Young	343719	11170334	0.99
2	Balls to the V	2	2	1		342562	5510424	0.99
3	Fast As a Sha	3	2	1	F. Baltes, S. K	230619	3990994	0.99
4	Restless and	3	2	1	F. Baltes, R.A	252051	4331779	0.99
5	Princess of th	3	2	1	Deaffy & R.A.	375418	6290521	0.99
6	Put The Finge	1	1	1	Angus Young	205662	6713451	0.99
7	Let's Get It U	1	1	1	Angus Young	233926	7636561	0.99
8	Inject The Ve	1	1	1	Angus Young	210834	6852860	0.99
9	Snowballed	1	1	1	Angus Young	203102	6599424	0.99
10	Evil Walks	1	1	1	Angus Young	263497	8611245	0.99
11	C.O.D.	1	1	1	Angus Young	199836	6566314	0.99
12	Breaking The	1	1	1	Angus Young	263288	8596840	0.99
13	Night Of The	1	1	1	Angus Young	205688	6706347	0.99
14	Spellbound	1	1	1	Angus Young	270863	8817038	0.99

id	title	artist_id
1	For Those About To Ro	1
2	Balls to the Wall	2
3	Restless and Wild	2
4	Let There Be Rock	1
5	Big Ones	3
6	Jagged Little Pill	4
7	Facelift	5
9	Plays Metallica By Fou	7
10	Audioslave	8
11	Out Of Exile	8
12	BackBeat Soundtrack	9
13	The Best Of Billy Cobha	10
14	Alcohol Fueled Brewta	11

id	name
1	AC/DC
2	Accept
3	Aerosmith
4	Alanis Morissette
5	Alice In Chains
7	Apocalyptica
8	Audioslave
9	BackBeat
10	Billy Cobham
11	Black Label Society
12	Black Sabbath
13	Body Count

What can we do if some values are missing?

- Delete features with mostly missing values (columns)
- Delete instances with missing features (rows)
  - If rare

- **Feature imputation** methods try to “fill in the blanks
- Variants:
  - replacing with a constant
    - the mean feature value (numerical)
    - the mode (categorical or ordinal)
    - “flag” missing values using out of range values
  - replacing with a random value
  - predicting the feature value from other features

sepal_lenght	sepal_width	petal_lenght	petal_width	Class
5.0	3.3	1.4	0.2	Iris-setosa
7.0	**NaN**	4.7	1.4	Iris-versicolor
5.7	2.8	4.1	1.3	
6.3	**NaN**	6.0	2.5	Iris-virginica

Data might not be “missing at random” or due to technical issues

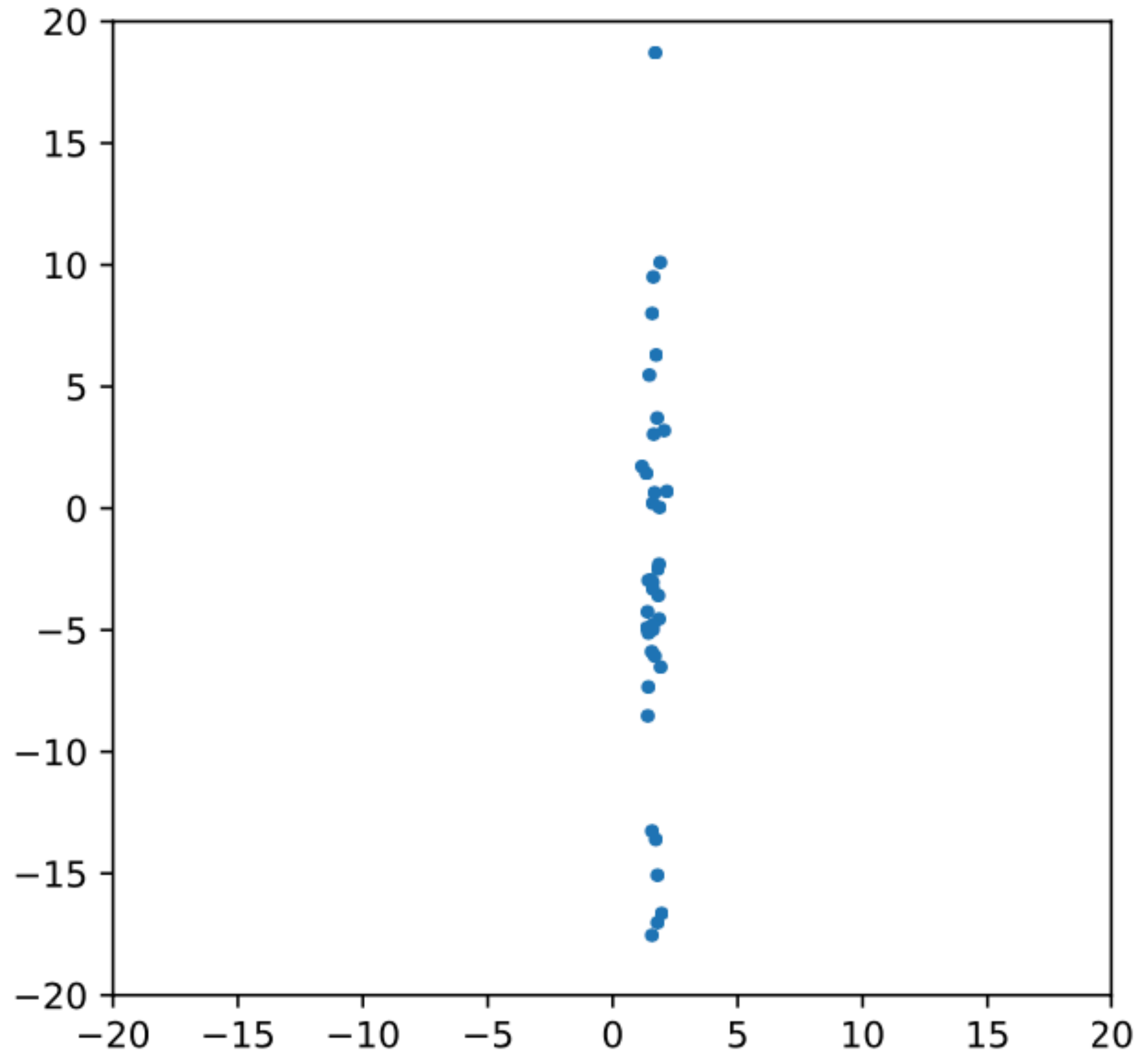
It might be meaningful that instances have missing features!

What if our features look like this?

- What if the features have different magnitudes?
- Does it matter if a feature is represented as meters or millimetres?
- What if there are outliers?

- Values spread strongly affects many models:
  - linear models (linear SVC, logistic regression, . . .)
  - neural networks
  - models based on distance or similarity (e.g. kNN)

- It does not matter for most tree-based predictors
  - they just consider thresholds of one feature at a time



Feature Normalisation

- Normalisation is needed for many algorithm to work propertly
  - Or to speed up training

- Min/Max scaling
  - Values scaled between 0 and 1

$$f_{new} = \frac{f - f_{min}}{f_{max} - f_{min}}$$

- Standard scaling
  - Rescales features to have zero mean and unit variance
  - Outliers can cause problems

$$f_{new} = \frac{f - \mu_f}{\sigma_f}$$

- Scaling to unit length (typically for document)

$$x_{new} = \frac{x}{|x|}$$

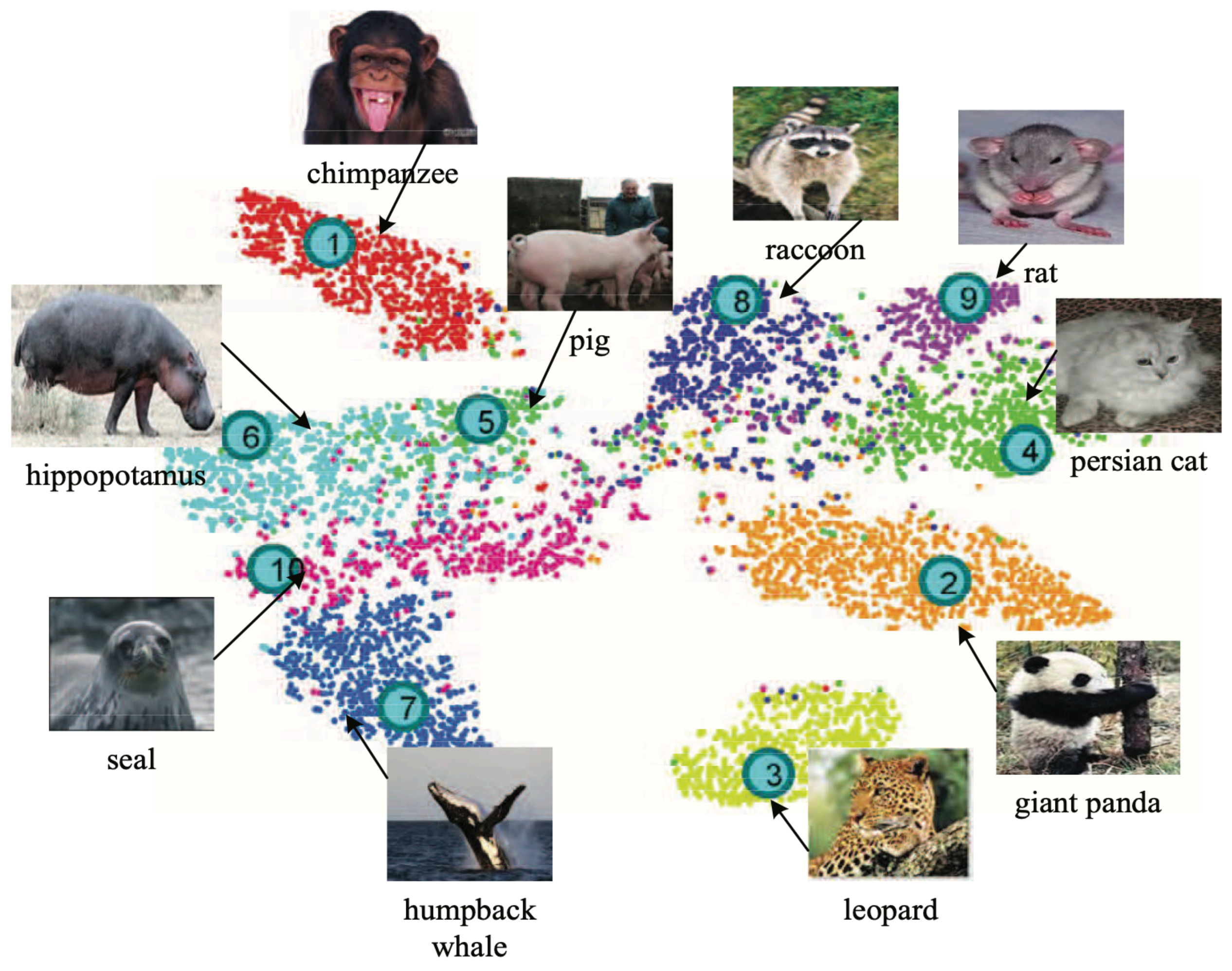
Other feature transformations

- we may try to improve performance by trying other transformations
  - logarithm, square root, . . .
  - TF-IDF
- Trial and error, exploration and your intuition

Feature Selection and Removal

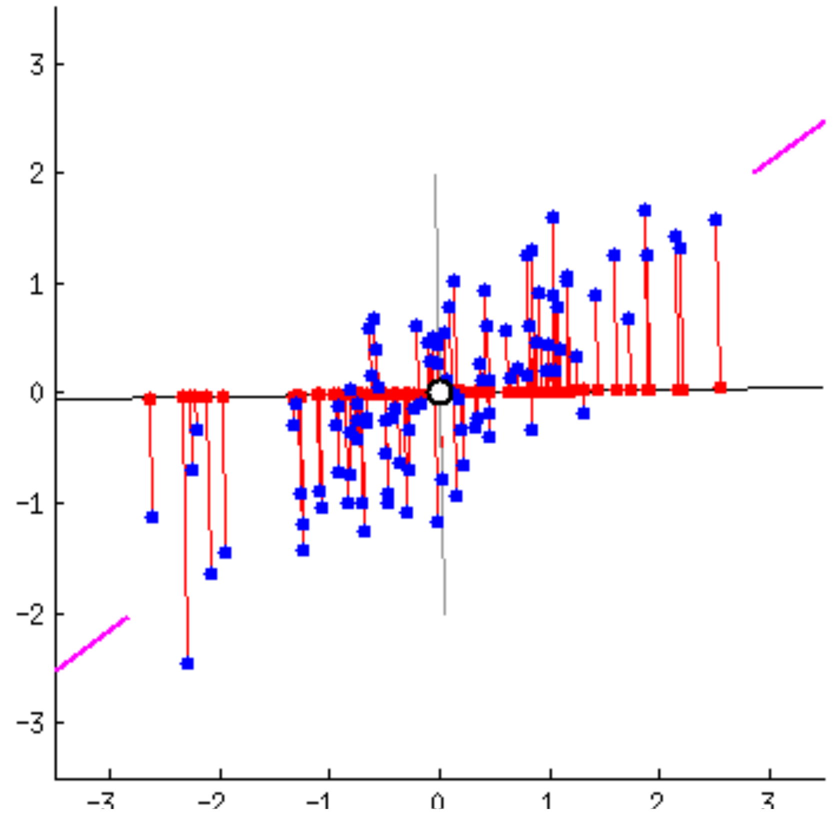
- In some cases, the number of features may be very large leading to several problems:
  - Important information is drowned out
  - Longer model training time
  - More complexity ⇒ bad for generalization
- Solution: leave out some features. But which ones?
  - Feature selection methods can find a useful subset
- **Idea**: find a subspace that retains most of the information about the original data
  - Pretty much as we were doing with Word embeddings
  - PRO: fewer dimensions make for datasets that are easier to explore and visualise, and faster training of ML algorithms
  - CONS: drop in prediction accuracy (less information)
- There are many different methods, *Principal Component Analysis* is a classic

Image from: https://arxiv.org/pdf/1703.08893.pdf

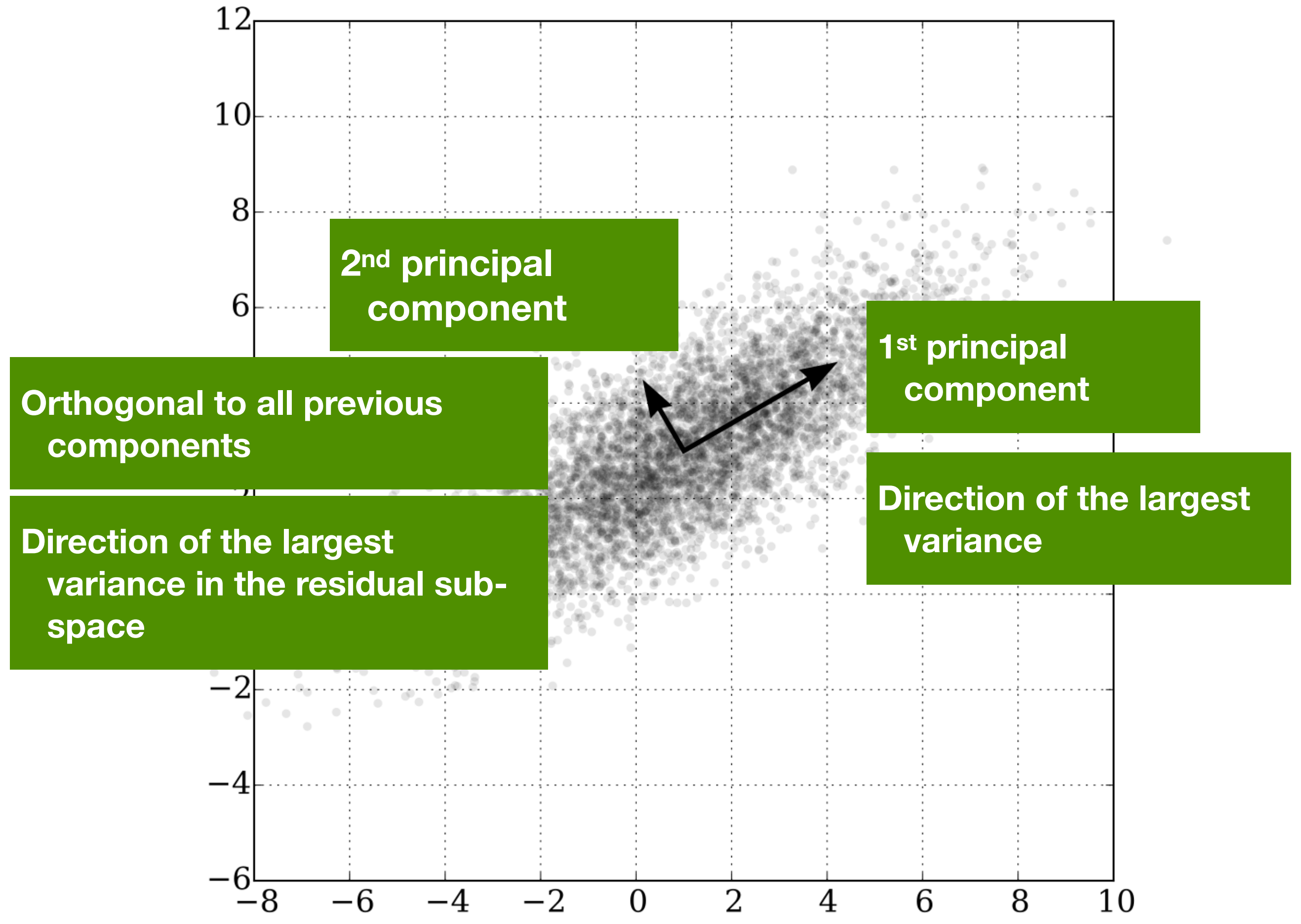


Principal Component Analysis

- Sometimes features are highly correlated with each other, therefore containing redundant information
- **Principal components** are new features that are constructed as linear combinations or mixtures of the initial features
  - Orthogonal projection of data onto lower-dimension linear space that:
    - maximizes the variance of projected data (purple line)
    - minimizes mean squared distance between data point and projections (sum of red lines)

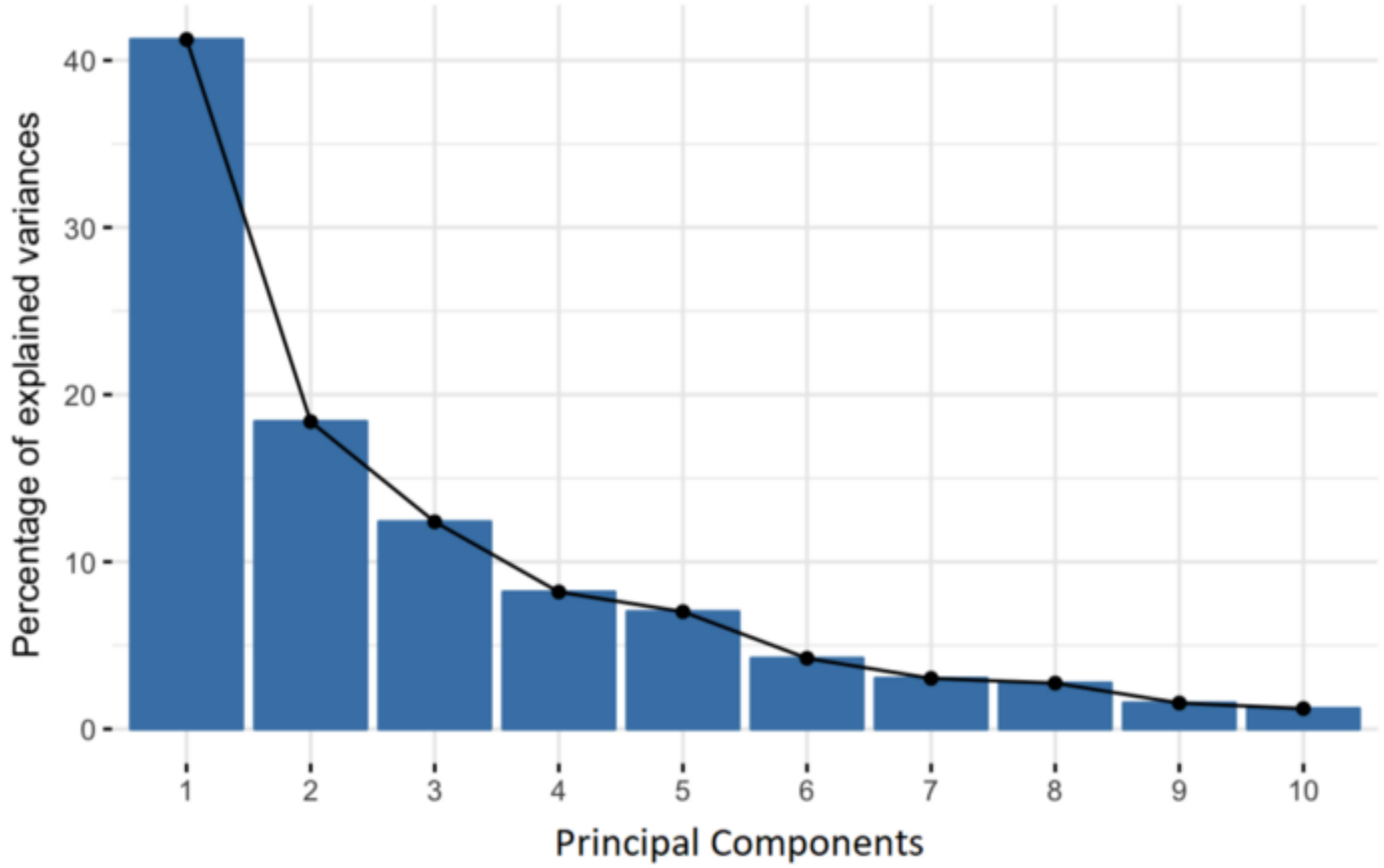


- The new features (i.e., principal components) are uncorrelated
  - Most of the information within the initial features is compressed into the first components



Dimensionality Reduction

- Use the PCA transformation of the data instead of the original features
  - PCA keeps most of the variance of the data
  - So, we are reducing the dataset to features that retain meaningful variations of the dataset

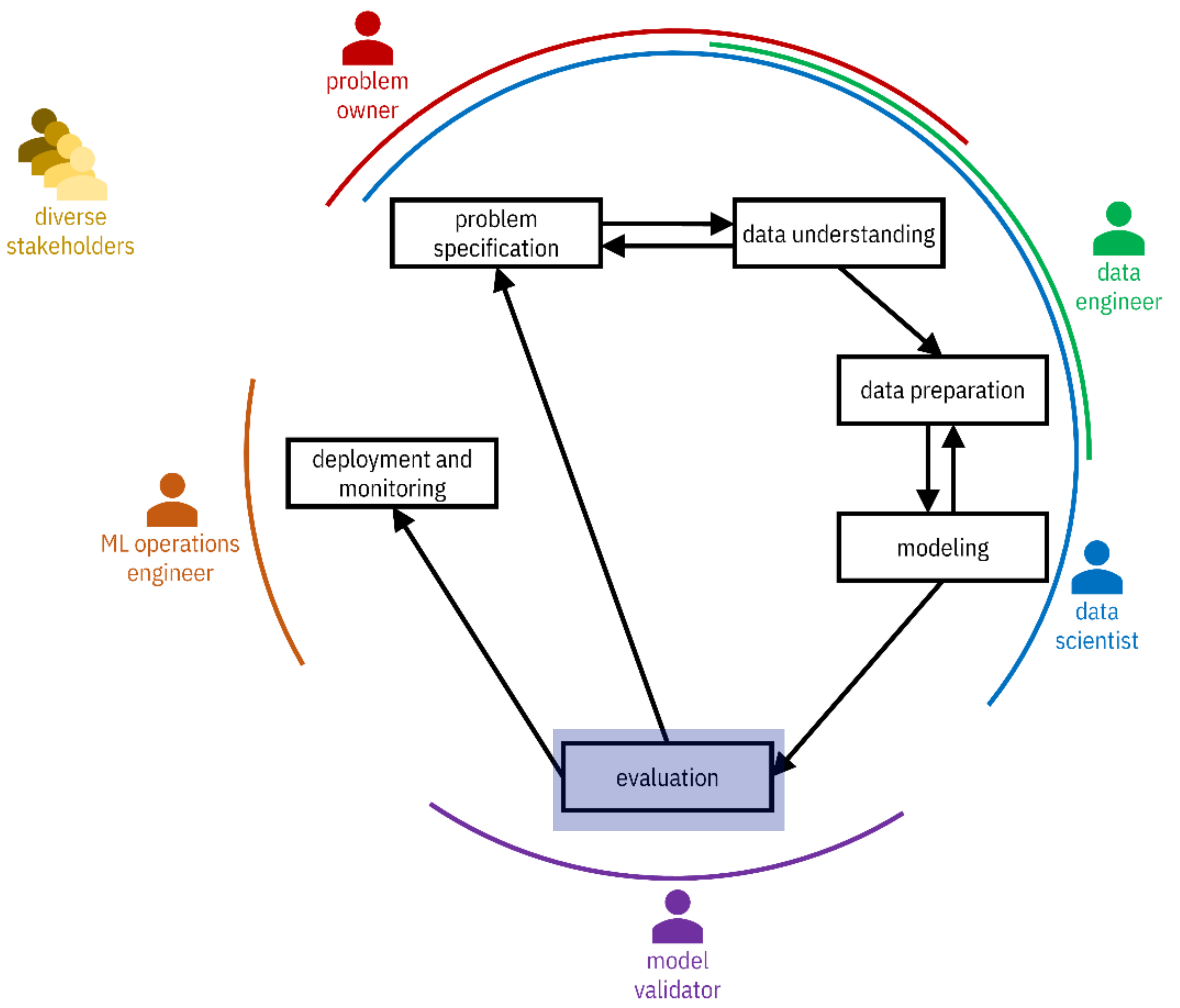


  - Ignore the components of less significance (e.g. only pick the first 3 components)

Evaluation

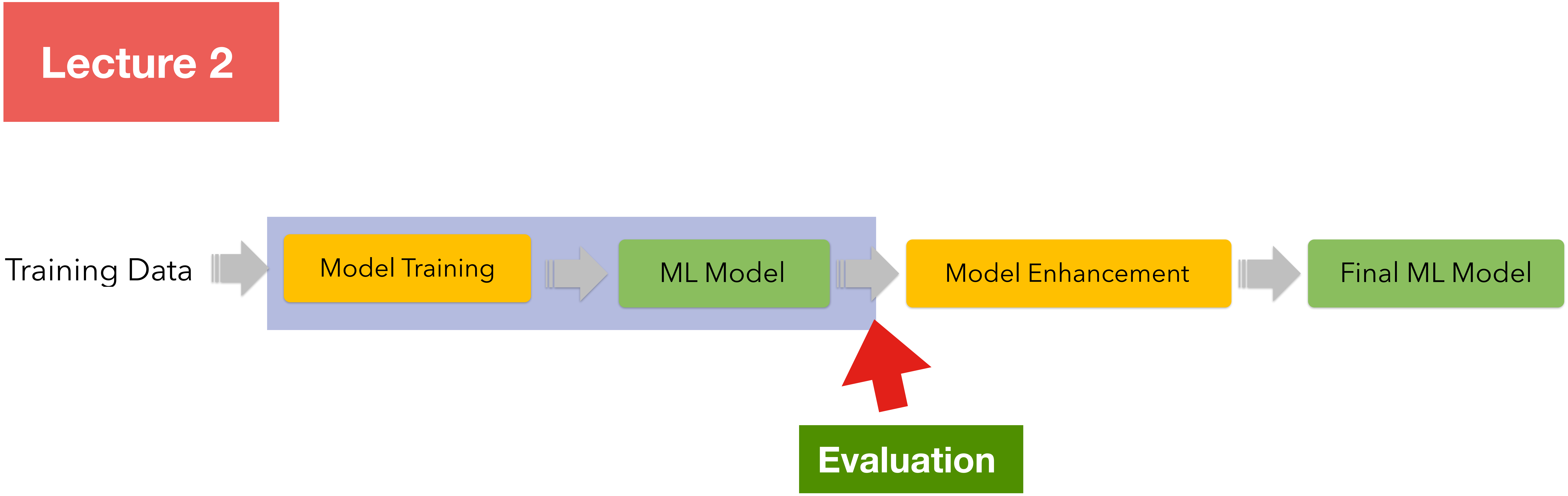
Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology

Lecture 2



https://www.the-modeling-agency.com/crisp-dm.pdf

How do machines learn?



How to Evaluate?

- Metric
  - How to measure errors?
- Training of Machine Learning algorithm
  - How to “help” the ML model to generalise?
- Experiment
  - How to pick the best ML model?

How to evaluate? - Metric

- Errors are almost inevitable!
  - How to measure errors?

- We’re generally interested in the following:
  - How often is the prediction wrong?
  - How is the prediction wrong?
  - What is the cost of wrong predictions?
  - How does the cost vary by the type of prediction that was wrong?
  - How can we minimize costs? (or regret?)

- Select an evaluation procedure (a “metric”)
  - **Ok, but which one?**

Classification

- **Accuracy**
  - In Classification, the model with highest accuracy is not necessarily the best model
  - Some errors (e.g. False Negative) may be much more expensive than others
    - Usually due to imbalanced trained datasets

$$Accuracy = \frac{\#CorrectPredictions}{\#Predictions}$$

- **Confusions Matrix**
  - Describes the complete performance of the model

True Positive

Actual Class

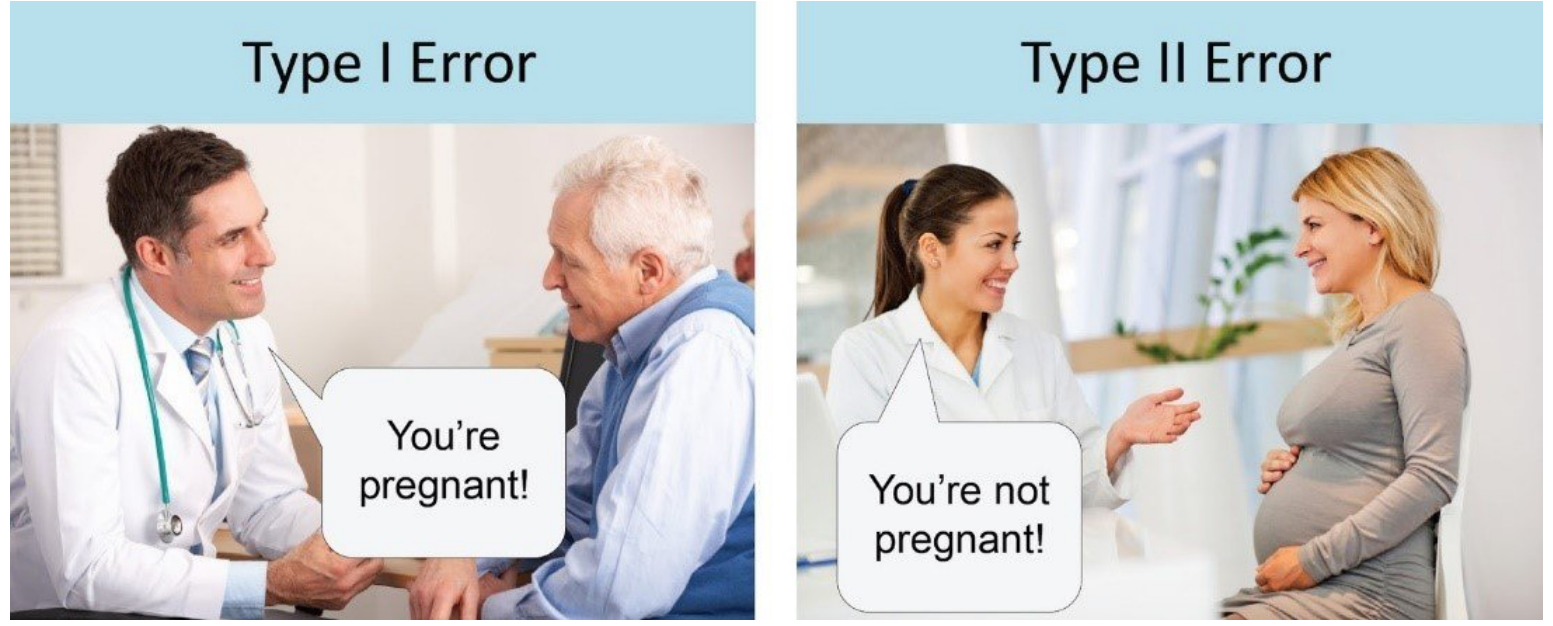
Predicted Class

	Yes	No
Yes	50	10
No	40	100

False Negative (**Type-1 Error**)

True Negative

False Positive (**Type-2Error**)



$$Accuracy = \frac{\#TruePositices + \#FalseNegatives}{\#AllPredictions}$$

All errors are not equal

- Depending on your task, different errors have different costs
- Cost of “false negative” in pregnancy detection? Cost of “false positive” in pregnancy detection?
- In law enforcement?
- In detecting the “Alexa” command?
- In detecting person on the road?

FALSE POSITIVES: SELF-DRIVING CARS AND THE AGONY OF KNOWING WHAT MATTERS

According to a preliminary report released by the National Transportation Safety Board last week, Uber’s system detected pedestrian Elaine Herzberg six seconds before striking and killing her. It identified her as an unknown object, then a vehicle, then finally a bicycle. (She was pushing a bike, so close enough.) About a second before the crash, the system determined it needed to slam on the brakes. But Uber hadn’t set up its system to act on that decision, the NTSB explained in the report. The engineers prevented their car from making that call on its own “to reduce the potential for erratic vehicle behavior.” (The company relied on the car’s human operator to avoid crashes, which is a whole separate problem.)

ARN MORE

Uber’s engineers decided not to let the car auto-brake because they were worried the system would overreact to things that were unimportant or not there at all. They were, in other words, very worried about false positives.

https://www.wired.com/story/self-driving-cars-uber-crash-false-positive-negative/

Classification

- **Sensitivity** (True positive rate)
  - probability of a positive classification, conditioned on being in the correct class

$$Sensitivity = \frac{TruePositive}{FalseNegative + TruePositive}$$

- **Specificity** (False positive rate)
  - probability of a negative classification, conditioned on not being in the correct class

$$Specificity = \frac{FalsePositive}{FalsePositive + TrueNegative}$$

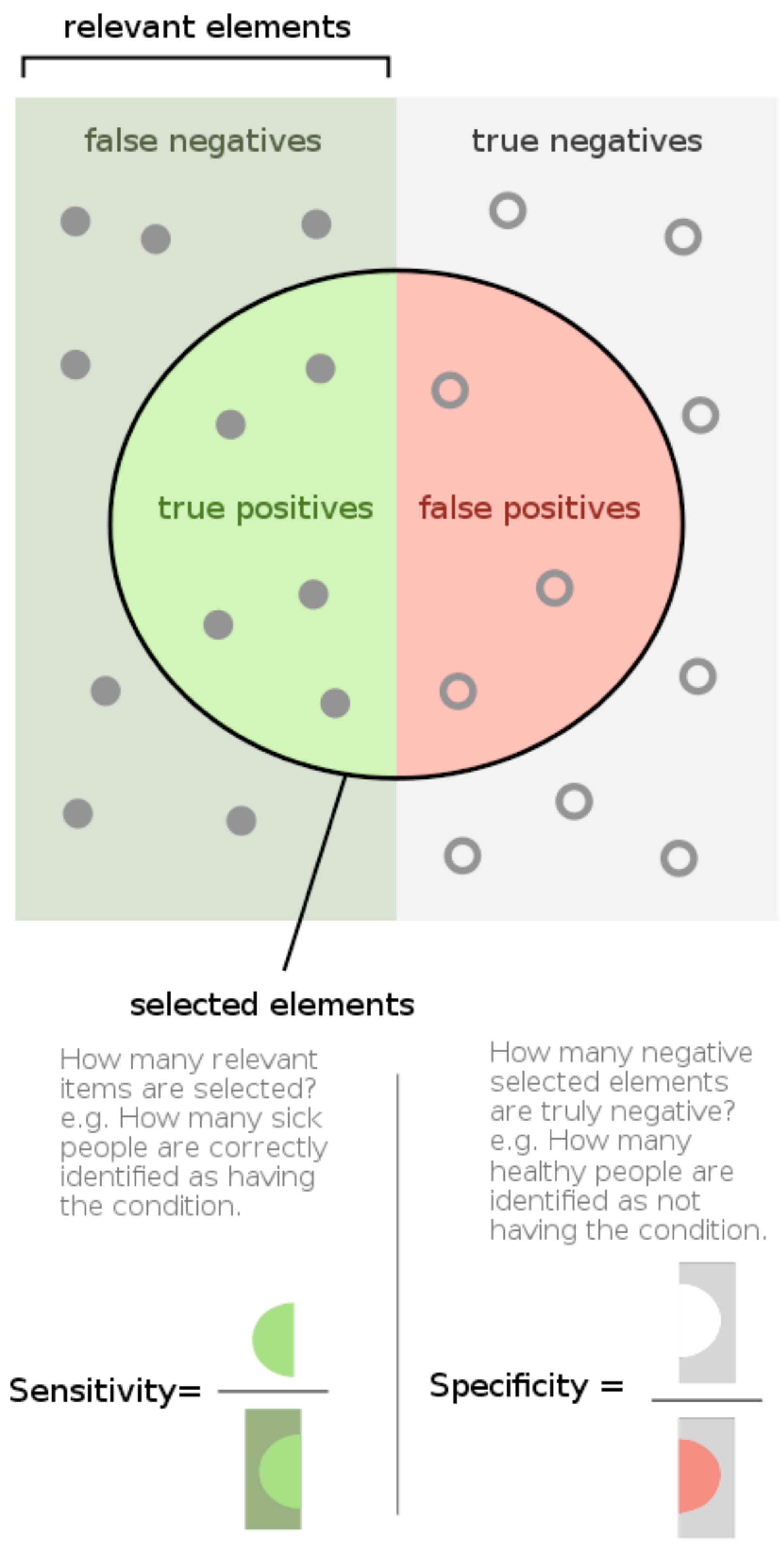
- **F1-Score**
  - Harmonic mean between **precision** (how many instances correctly classified), and **recall** (how many relevant instance are correctly classified)

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

  - What is the implicit assumption about the costs of errors?

$$F_1 = 2 * \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}}$$



https://upload.wikimedia.org/wikipedia/commons/thumb/5/5a/Sensitivity_and_specificity_1.01.svg/512px-Sensitivity_and_specificity_1.01.svg.png

Choosing Metrics

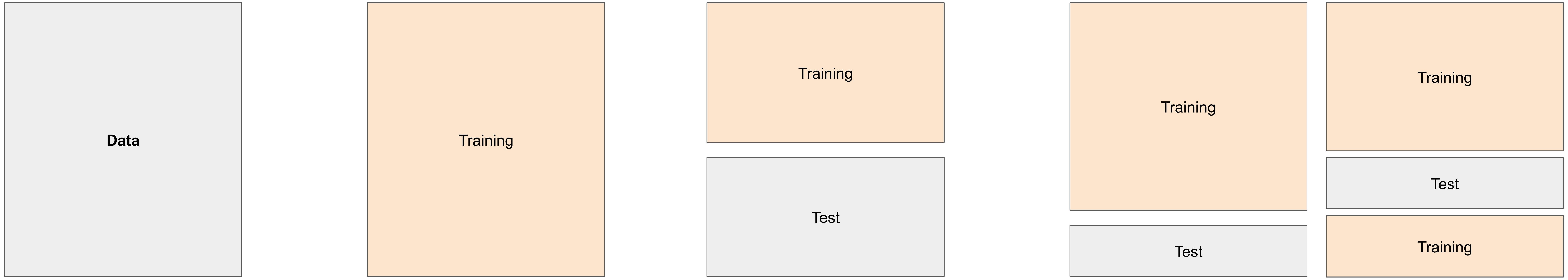
- If a high precision is a hard constraint, do the best recall
  - search engine results, grammar correction: Intolerant to FP
  - Metric: Recall at Precision = XX %

- If a high recall is a hard constraint, do best precision
  - medical diagnosis: Intolerant to FN
  - Metric: Precision at Recall = 100 %

- Capacity constrained (by K)
  - Metric: Precision in top-K

How to evaluate? - Training

- Apply your model to a held-out test set and evaluate
  - the test set must be different from the training set

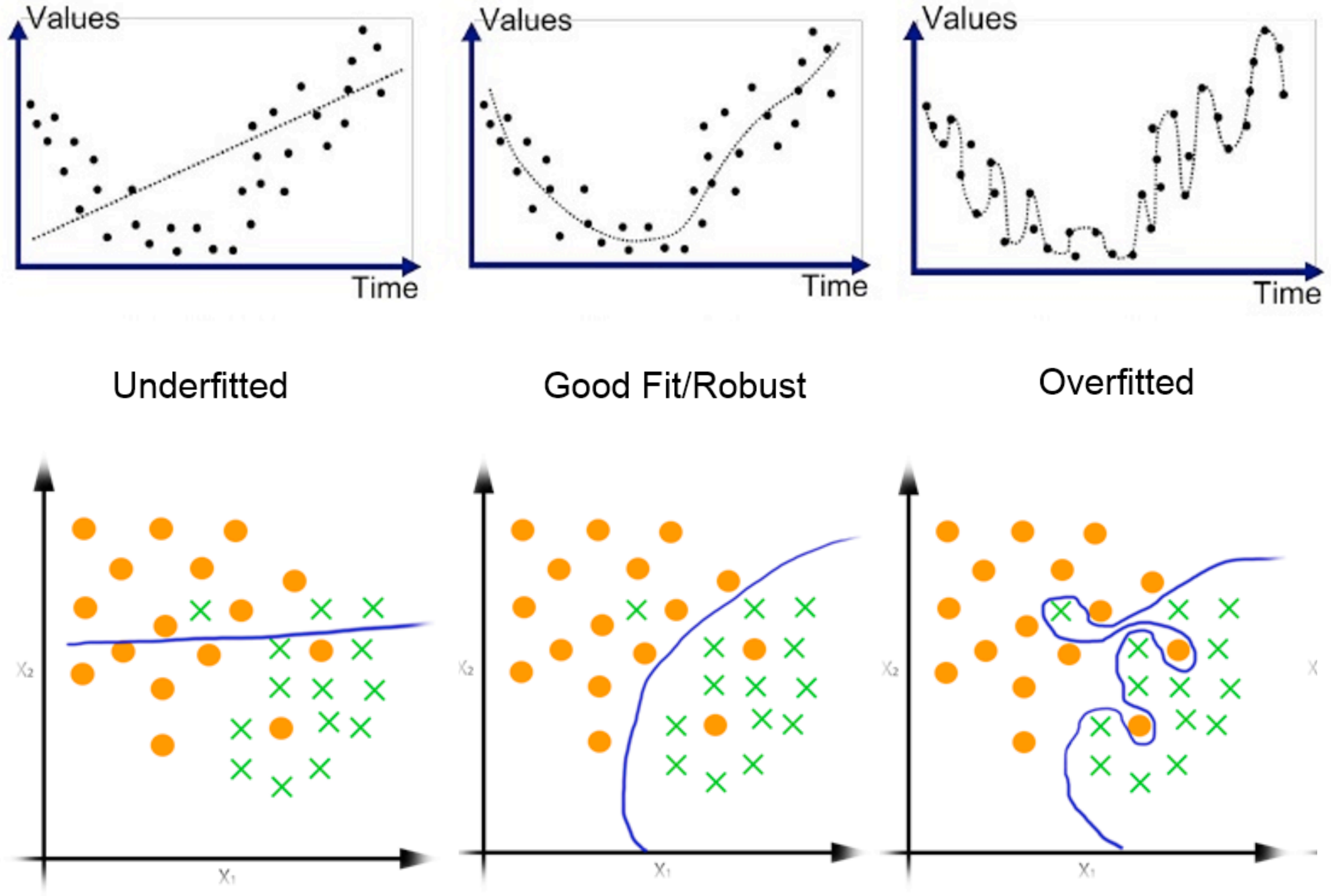


No free-lunch

- There is no one best machine learning algorithm for all problems and datasets

- **Generalisation**
  - How well does a learned model generalize from the data it was trained on to a new evaluation set?

Regression

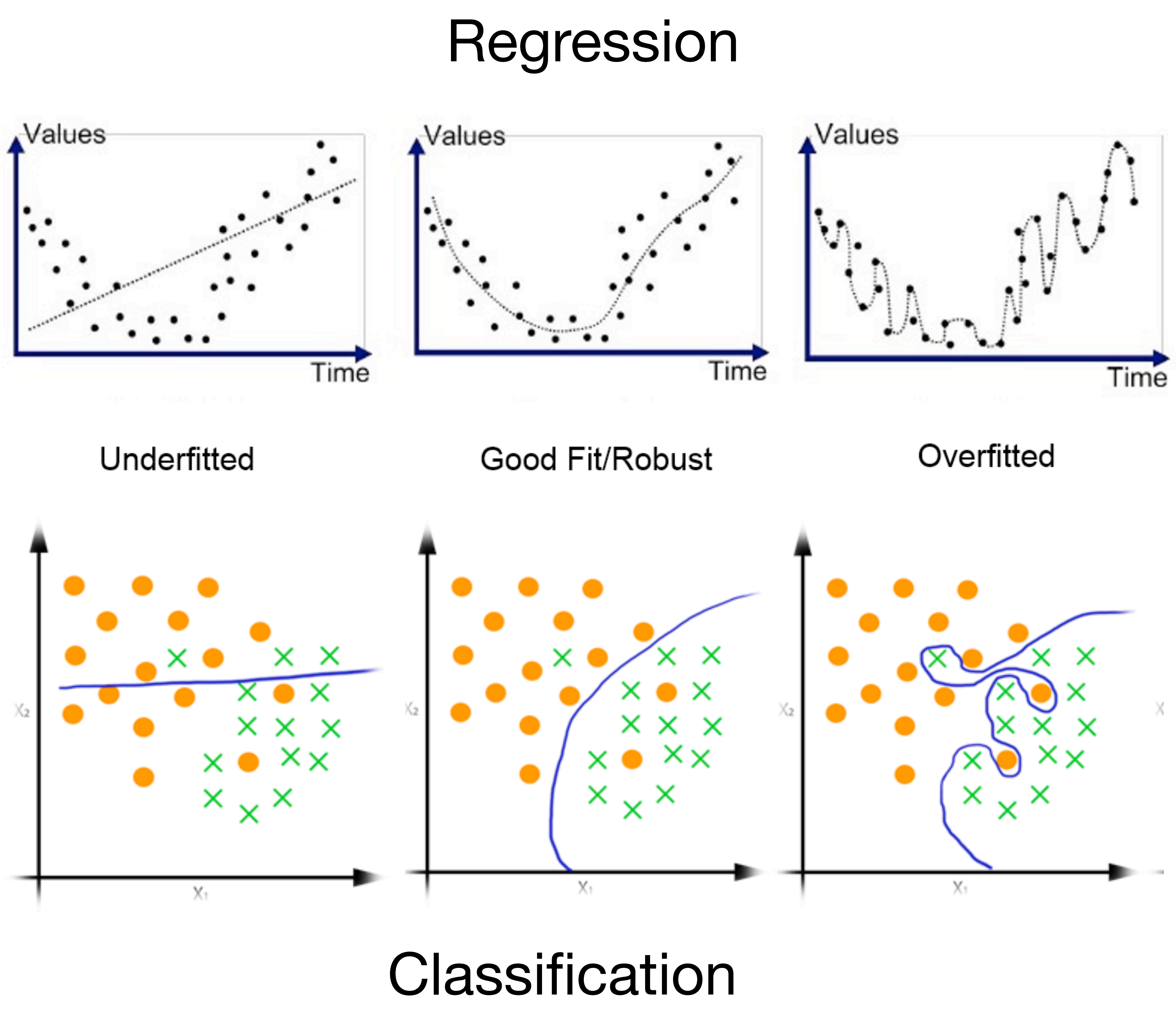


Classification

Generalisation

- **Challenge**: achieving good generalization and a small error rate
- Components of expected loss
  - **Noise** in our observations: **unavoidable**
  - **Bias**: how much the average model over all training sets differs from the true model
    - Error due to inaccurate assumptions/simplifications made by the model
  - **Variance**: how much models estimated from different training sets differ from each other

- Protect against **overfitting**
  - learning a model that too closely matches the idiosyncrasies of the training data
  - model is too “complex” and fits irrelevant characteristics (noise) in the data
    - Low bias and high variance
    - Low training error and high test error
- Protect against **underfitting**
  - learning a model that does not adequately capture the patterns in the training data
  - The model is too “simple” to represent all the relevant class characteristics
    - High bias and low variance
    - High training error and high test error



How to evaluate? - Experiments

- Compare to one or more baselines
  - existing solution
  - trivial (new) solution
  - rule-based solution
  - Multiple ML models

Admin

Week 6 Tasks

- Submit questions for Week 6
  - https://forms.office.com/r/h7KwSwGR0c
- Feel free to submit new questions for the previous weeks!
- Give us your feedback about Module 2
  - https://forms.office.com/r/G2rAazNHkG
- Prepare for Friday’s tutorial and individual assignment!
  - This one is a bit more intensive - so, please reserve the time!

Machine Learning For Design

Lecture 6 - Machine Learning and Natural Language Processing / Part 2

Alessandro Bozzon

09/03/2022

mlfd-io@tudelft.nl
www.ml4design.com

Credits

- CIS 419/519 Applied Machine Learning. Eric Eaton, Dinesh Jayaraman. https://www.seas.upenn.edu/~cis519/spring2020/
- EECS498: Conversational AI. Kevin Leach. https://dijkstra.eecs.umich.edu/eecs498/
- CS 4650/7650: Natural Language Processing. Diyi Yang. https://www.cc.gatech.edu/classes/AY2020/cs7650_spring/
- Natural Language Processing. Alan W Black and David Mortensen. http://demo.clab.cs.cmu.edu/NLP/
- IN4325 Information Retrieval. Jie Yang.
- Speech and Language Processing, An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Third Edition. Daniel Jurafsky, James H. Martin.
- Natural Language Processing, Jacob Eisenstein, 2018.
- A Step-by-Step Explanation of Principal Component Analysis (PCA). https://builtin.com/data-science/step-step-explanation-principal-component-analysis