https://github.com/saubury/cat-predict

Can ML predict where my cat is now?

Data & Analytics Wednesday
8th June, 2022

Simon Aubury

/thoughtworks

@Simon Aubury

Simon Aubury

Principal Data Engineer

/thoughtworks

@Simon Aubury

Could I train a model to predict where Snowy would go throughout her day?

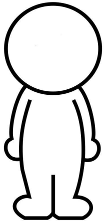
Part 1 - ML Bootcamp

/thoughtworks

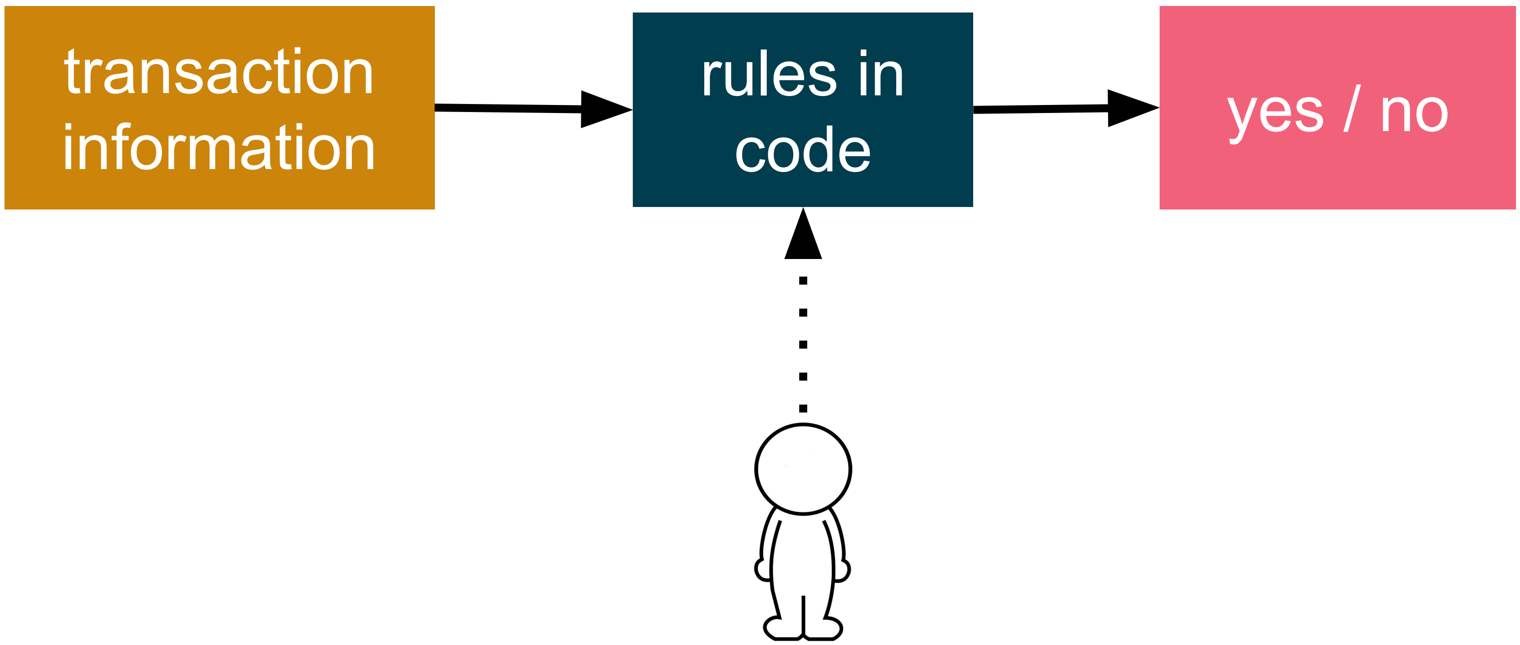
@Simon Aubury

Data to insight

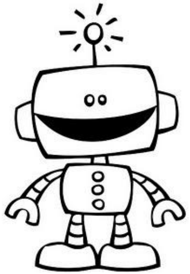
Data to insight



Data to insight



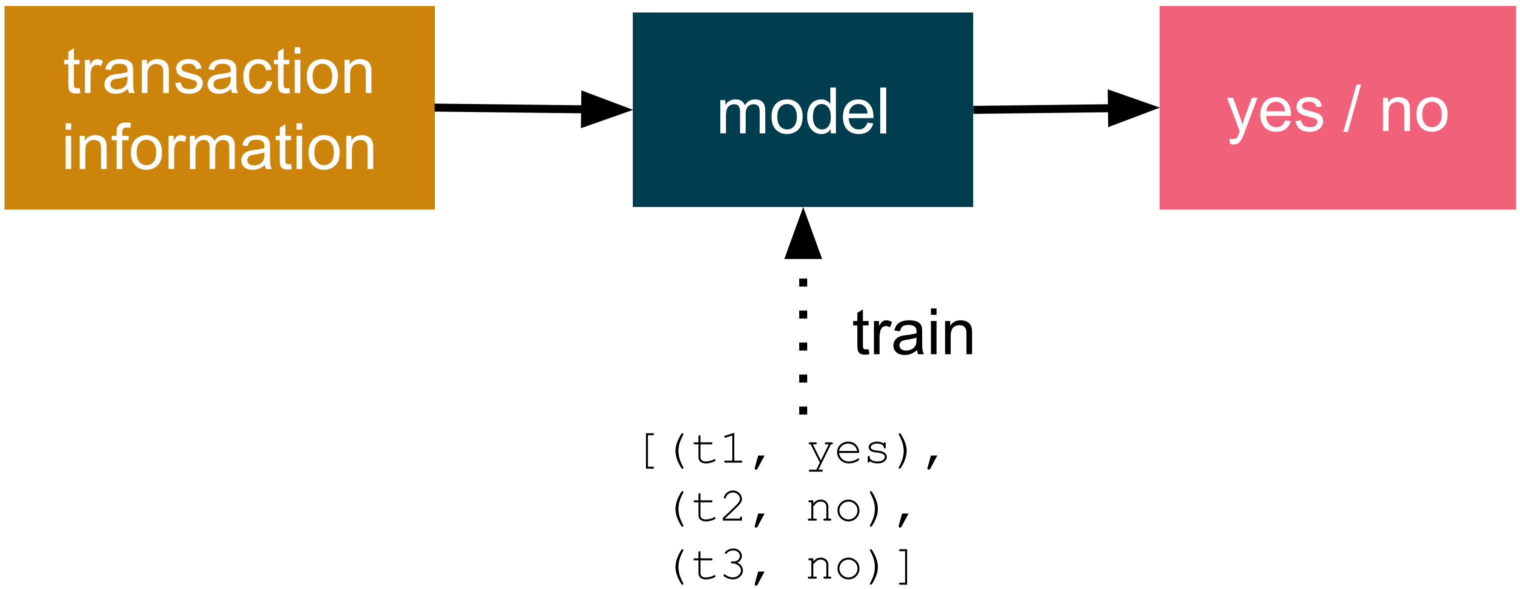
Data to insight



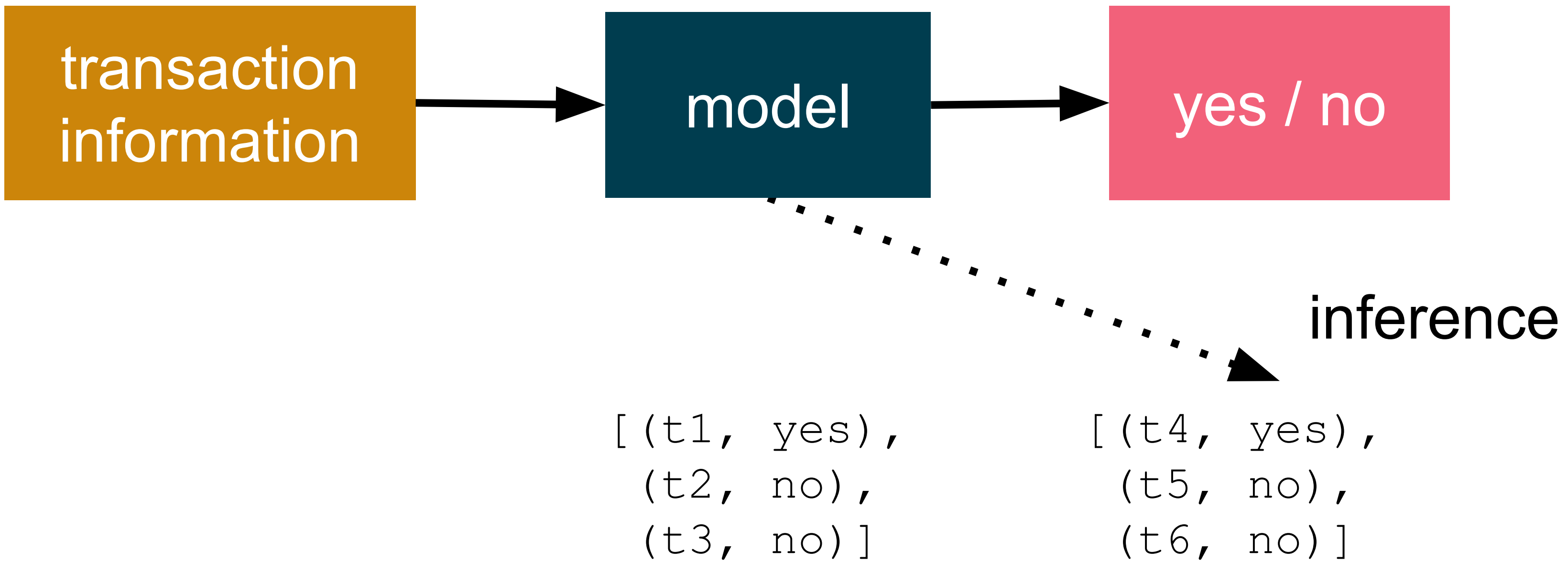
Data to insight


```
[(t1, yes),
 (t2, no),
 (t3, no),
 ...]
```

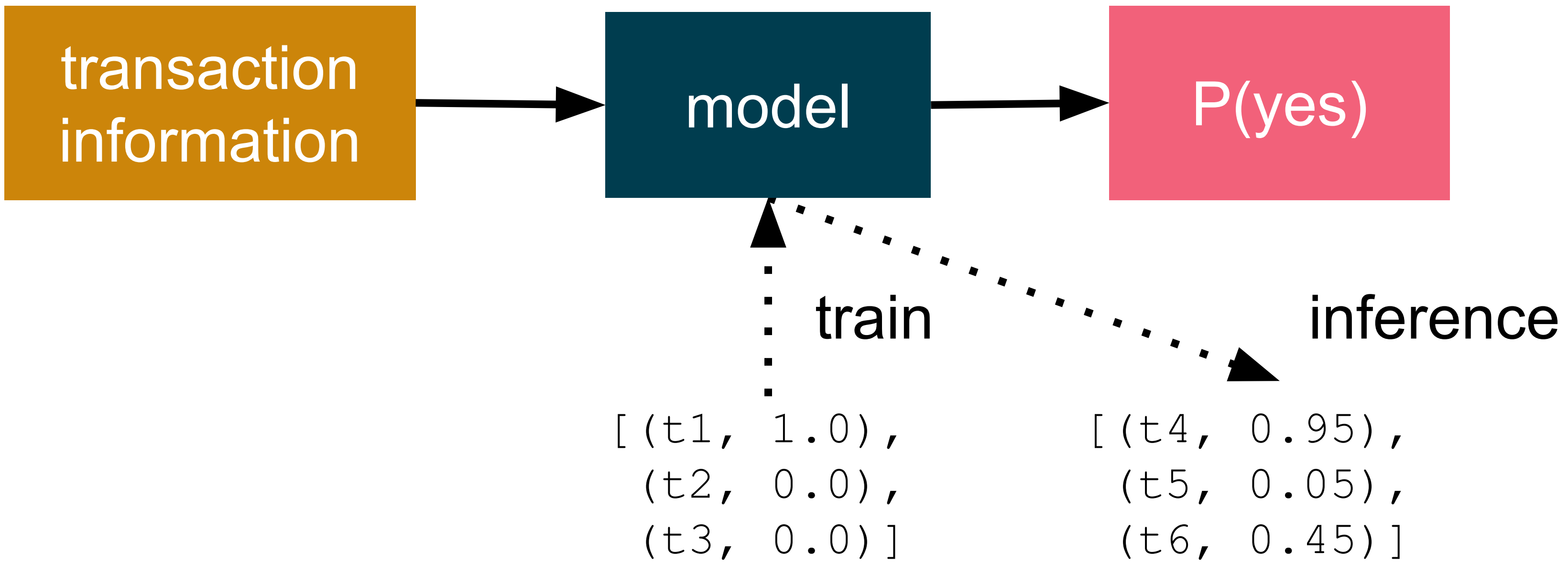
Data to insight



Data to insight



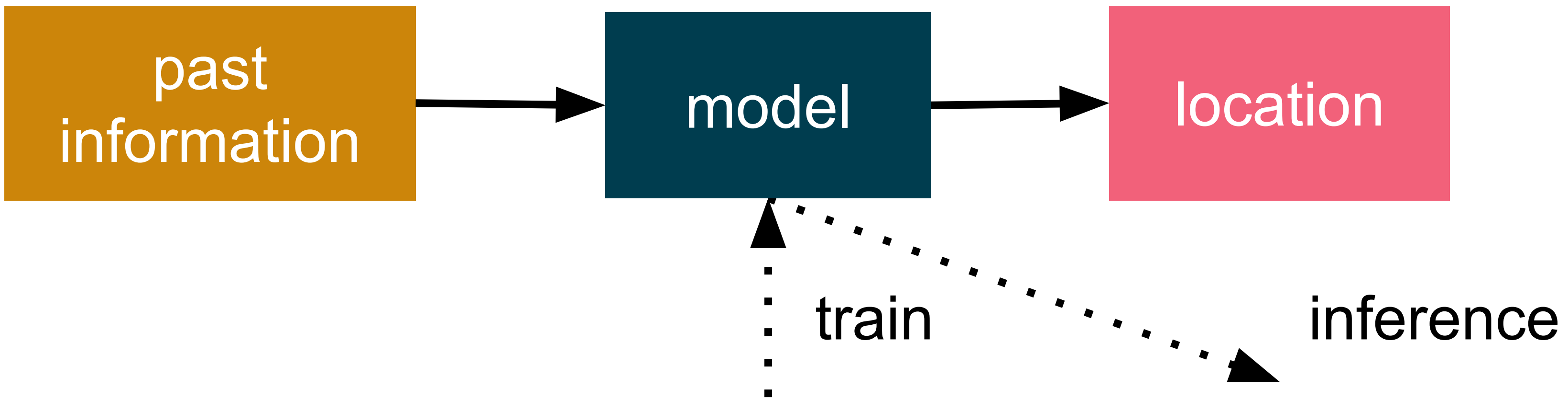
Data to insight



Cats

What does this mean?

Cat location prediction




```
9am, cold, study          8am, cold, study(.95)
9am, hot, garden          10am, hot, garden(.80)
6pm, hot, dining          6pm, cold, study (.62)
```

How to approach most ML problems in 7 steps

Define problem	Predict cat location
Obtain, load and explore data	Some cool hardware
Prepare and clean data for modelling	Data transformation with dbt
Train model	Scikit-learn, Random forest classifier
Evaluate model	Feature importance score; visualize decision tree
Tune / improve model	Tune parameters and feature engineering
Use the model	Streamlit app

Iterate!

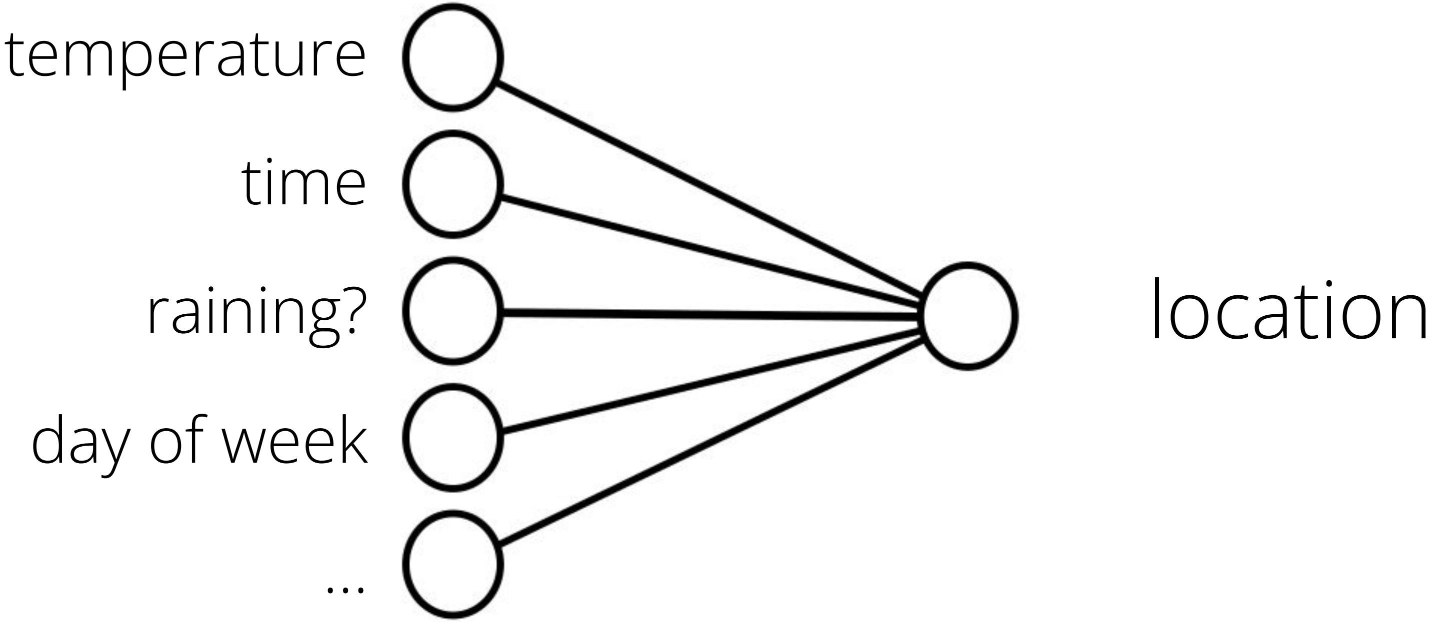
@Simon Aubury

How to approach most ML problems in 7 steps

Define problem	Predict cat location
Obtain, load and explore data	Some cool hardware
Prepare and clean data for modelling	Data transformation with dbt
Train model	Scikit-learn, Random forest classifier
Evaluate model	Feature importance score; visualize decision tree
Tune / improve model	Tune parameters and feature engineering
Use the model	Streamlit app

Iterate!

What data are we going to need?



Part 2 - Collect data & build model

/thoughtworks

@Simon Aubury

How to approach most ML problems in 7 steps

Define problem	Predict cat location
Obtain, load and explore data	Some cool hardware
Prepare and clean data for modelling	Data transformation with dbt
Train model	Scikit-learn, Random forest classifier
Evaluate model	Feature importance score; visualize decision tree
Tune / improve model	Tune parameters and feature engineering
Use the model	Streamlit app

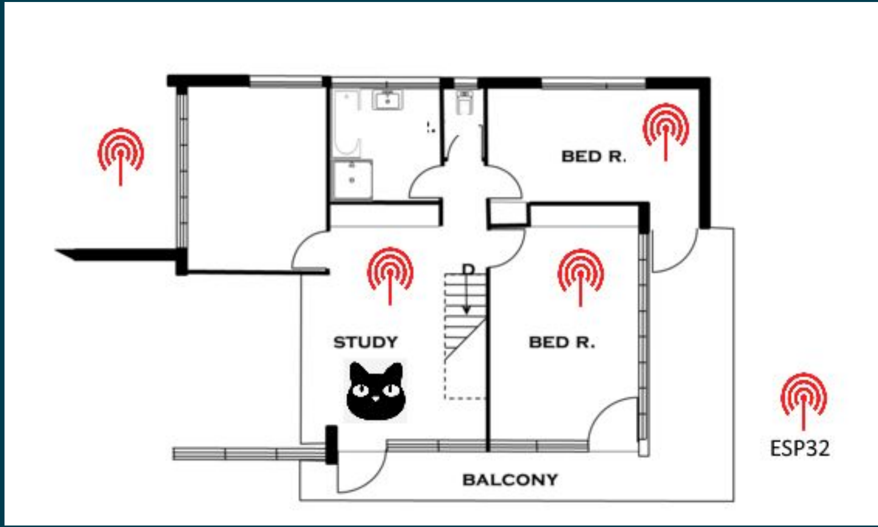
Iterate!

@Simon Aubury

Hardware for room level cat tracking

Snowy wears a “Tile” — a small, battery powered bluetooth transmitter.

Eight stationary ESP32 receivers to listen for the BLE Tile signal.



Hardware for environment logging

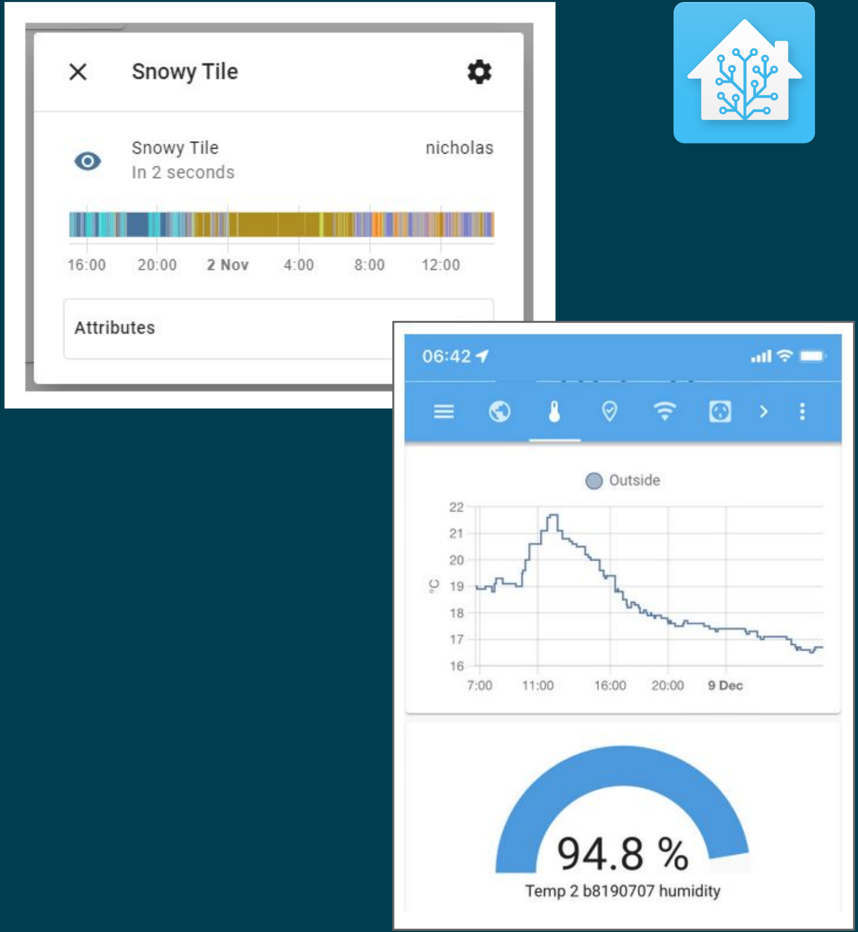
Xiaomi Temperature and Humidity Sensor communicate over large distances via the Zigbee wireless mesh network.

Placed four sensors in the house and two in external locations to capture outside conditions

Data collection platform

Home Assistant container running on home server

- Temperature and humidity measurements via Xiaomi integration
- ESPresense for location monitoring on MQTT topic
- SQLite replaced with Postgres
    - 6 months of retention
    - 18,000 updates a day **per sensor**



How to approach most ML problems in 7 steps

Define problem	Predict cat location
Obtain, load and explore data	Some cool hardware
Prepare and clean data for modelling	Data transformation with dbt
Train model	Scikit-learn, Random forest classifier
Evaluate model	Feature importance score; visualize decision tree
Tune / improve model	Tune parameters and feature engineering
Use the model	Streamlit app

Iterate!

@Simon Aubury

Summarising data

Lots of stuff

- Home Assistant stores all sensor updates in the “states” table
- Records sensors as they respond
  - 18,000 inserts a day per sensor
  - 120,000 inserts a day for useful sensors
  - Everything as time sequenced updates
- Goal is to summarise the data into hourly updates

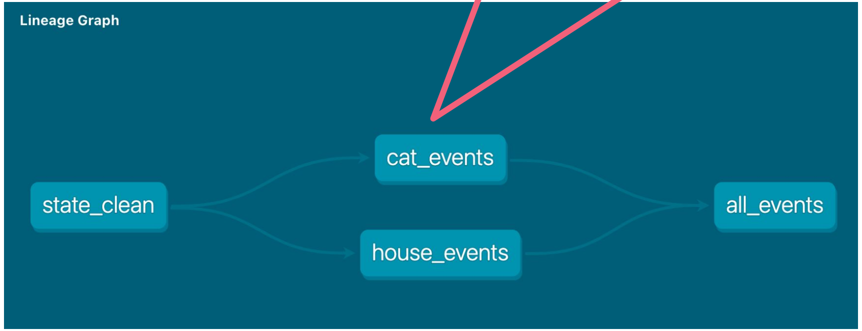
entity_id	created	attributes
sensor.snowy_tile	2021-12-08 17:12:03.634 +1100	{"distance":1.64,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:12:08.607 +1100	{"distance":1.59,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:12:13.628 +1100	{"distance":1.5,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:12:23.377 +1100	{"distance":1.4,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:12:29.489 +1100	{"distance":1.43,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:12:37.412 +1100	{"distance":1.45,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:12:42.406 +1100	{"distance":1.4,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:12:45.408 +1100	{"distance":1.25,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:12:50.489 +1100	{"distance":1.22,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:13:07.492 +1100	{"distance":1.25,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:13:17.446 +1100	{"distance":1.32,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:13:22.449 +1100	{"distance":1.4,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:13:27.557 +1100	{"distance":1.48,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:13:31.563 +1100	{"distance":1.58,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:13:36.054 +1100	{"distance":1.75,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:13:45.584 +1100	{"distance":1.83,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:13:47.623 +1100	{"distance":2.39,"friendly_name":"Snowy Tile"}
sensor.snowy_tile	2021-12-08 17:13:51.627 +1100	{"distance":2.22,"friendly_name":"Snowy Tile"}

created_local_hr	day_of_week	hr_of_day	indoor_temp	outside_temp	outside_humidity	cat_location
30/10/21 4:00	Saturday	04	24	18	44	study
30/10/21 5:00	Saturday	05	24	18	45	dining
30/10/21 6:00	Saturday	06	24	17	48	outside
30/10/21 7:00	Saturday	07	23	18	54	bedroom
30/10/21 8:00	Saturday	08	23	20	53	bedroom
30/10/21 9:00	Saturday	09	23	20	50	dining
30/10/21 10:00	Saturday	10	23	21	46	dining
30/10/21 11:00	Saturday	11	23	23	42	dining
30/10/21 12:00	Saturday	12	23	24	37	winter_garden
30/10/21 13:00	Saturday	13	24	24	37	study
30/10/21 14:00	Saturday	14	24	23	38	study
30/10/21 15:00	Saturday	15	24	22	40	study
30/10/21 16:00	Saturday	16	24	21	42	dining

dbt - let's SQL it …

dbt_


```
{{ config(materialized='view') }}

with cte AS
(
    select created_local_hr
    , state
    , count(*)
    , ROW_NUMBER() OVER (PARTITION BY created_local_hr ORDER BY COUNT(*) DESC) rn
    from {{ ref('state_clean') }}
    where entity_id = 'sensor.snowy_tile'
    group by created_local_hr, state
)
--
select created_local_hr, state as cat_location
from cte
where rn=1
```



How to approach most ML problems in 7 steps

Define problem	Predict cat location
Obtain, load and explore data	Some cool hardware
Prepare and clean data for modelling	Data transformation with dbt
Train model	Scikit-learn, Random forest classifier
Evaluate model	Feature importance score; visualize decision tree
Tune / improve model	Tune parameters and feature engineering
Use the model	Streamlit app

Model - Random forest decision tree

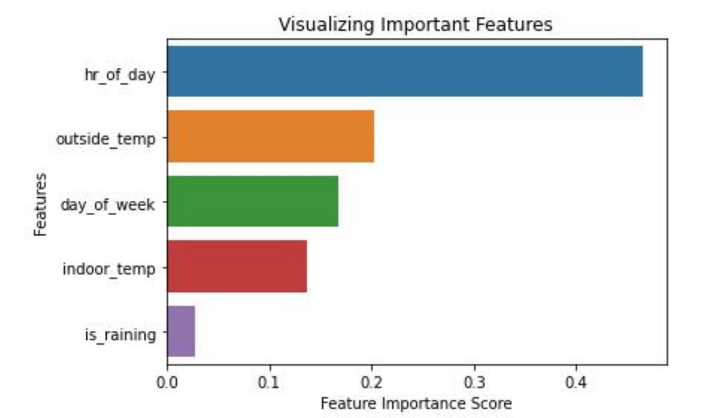
Scikit-learn, Logistic Regression

Derived feature

Derived feature

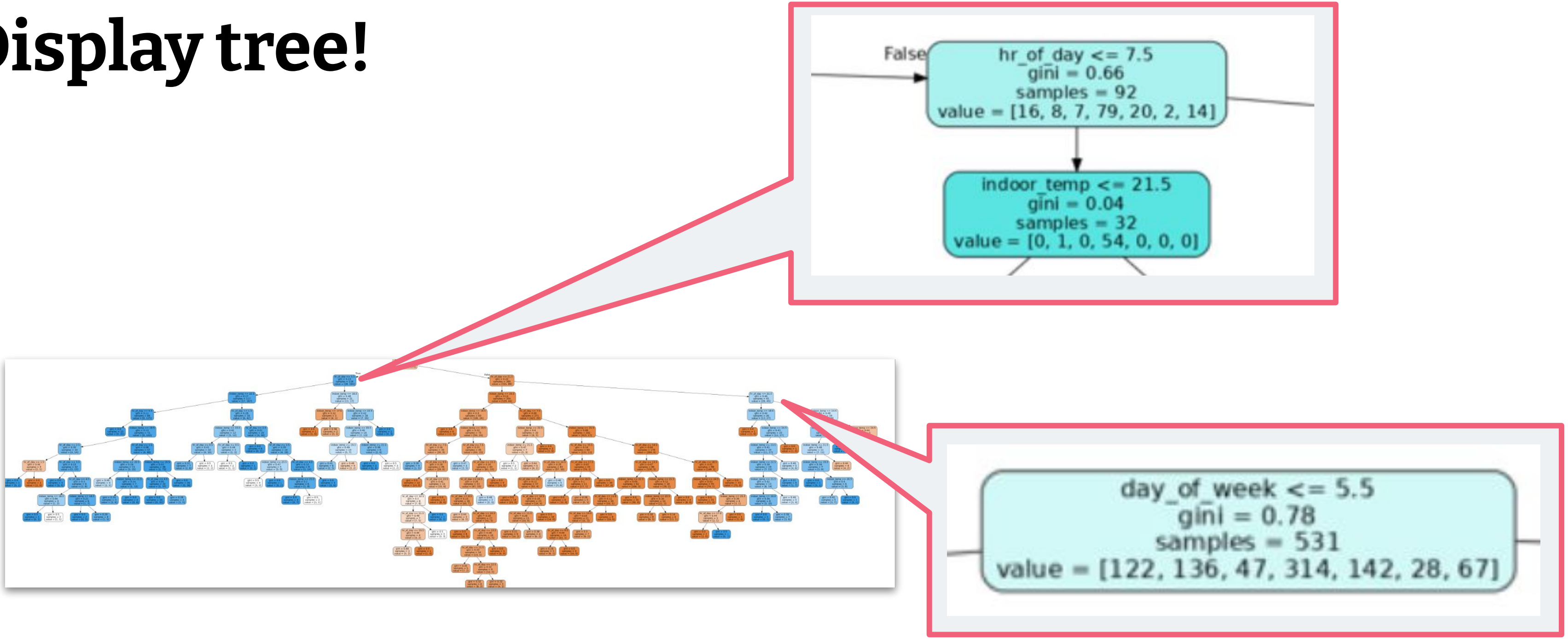
created_local_hr	day_of_week	hr_of_day	indoor_temp	outside_temp	outside_humidity	is_raining	cat_location
30/10/21 4:00	Saturday	04	24	18	44	FALSE	study
30/10/21 5:00	Saturday	05	24	18	45	FALSE	dining
30/10/21 6:00	Saturday	06	24	17	48	FALSE	outside
30/10/21 7:00	Saturday	07	23	18	54	FALSE	bedroom
30/10/21 8:00	Saturday	08	23	20	53	FALSE	bedroom
30/10/21 9:00	Saturday	09	23	20	50	FALSE	dining
30/10/21 10:00	Saturday	10	23	21	46	FALSE	dining
30/10/21 11:00	Saturday	11	23	23	42	FALSE	dining
30/10/21 12:00	Saturday	12	23	24	37	FALSE	winter_garden
30/10/21 13:00	Saturday	13	24	24	37	FALSE	study
30/10/21 14:00	Saturday	14	24	23	38	FALSE	study
30/10/21 15:00	Saturday	15	24	22	40	FALSE	study
30/10/21 16:00	Saturday	16	24	21	42	FALSE	dining
30/10/21 17:00	Saturday	17	23	20	45	FALSE	outside
30/10/21 18:00	Saturday	18	23	19	47	FALSE	outside
30/10/21 19:00	Saturday	19	23	18	47	FALSE	bedroom

```python
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
import pickle
import numpy as np

cat_df = pd.read_csv('./cat_events.csv')
# cat_df.head()

# extract just the hour
cat_df['hr_of_day'] = cat_df['created_local_hr'].str[11:13].astype(int)
cat_df['day_of_week'] = pd.to_datetime(cat_df['created_local_hr']).dt.dayofweek.astype
cat_df.drop('created_local_hr', axis=1, inplace=True)
cat_df.dropna(axis=0, how='any', thresh=None, subset=None, inplace=True)

cat_df['is_raining'] = False
# DataFrame.where replace values where the condition is *False*. Read this as "when ou
cat_df['is_raining'].where(cat_df['outside_humidity'] < 90.0, True, inplace=True)
                                                                      Clear Out

# Drop outside_humidity now we have is_raining
cat_df.drop('outside_humidity', axis=1, inplace=True)

# Add a simple classification - is-nicho
cat_df['is_nicho'] = 'Yes'
cat_df['is_nicho'].where(cat_df['cat_location'] == 'nicholas', 'No', inplace=True)

cat_df.head(20)
```



@Simon Aubury

Display tree!



How to approach most ML problems in 7 steps

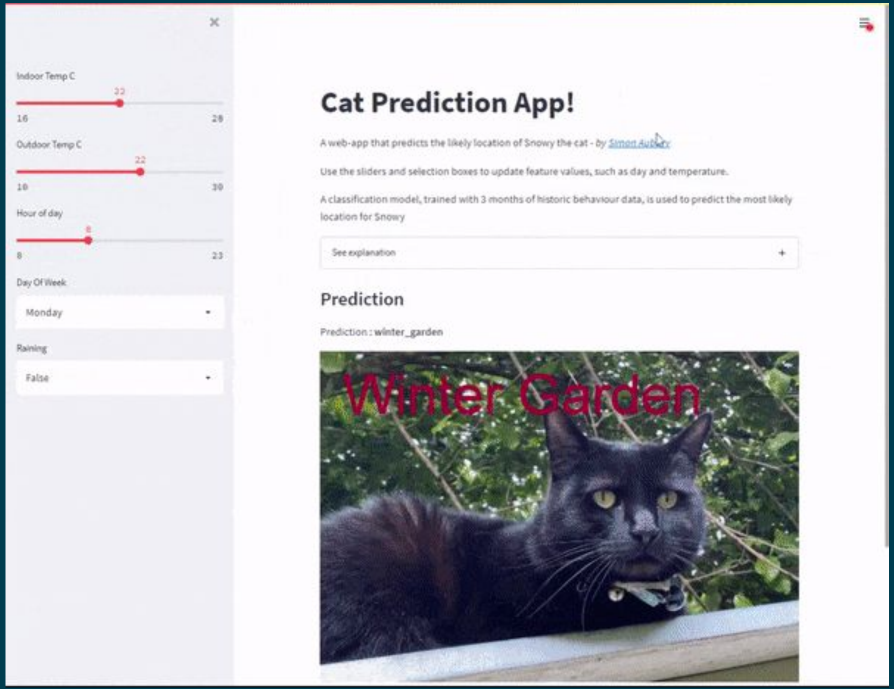
Define problem	Predict cat location
Obtain, load and explore data	Some cool hardware
Prepare and clean data for modelling	Data transformation with dbt
Train model	Scikit-learn, Random forest classifier
Evaluate model	Feature importance score; visualize decision tree
Tune / improve model	Tune parameters and feature engineering
Use the model	**Streamlit app**

@Simon Aubury

Streamlit App

You can try this yourself

https://cat-predict-app.herokuapp.com



What I've learnt & what’s next?

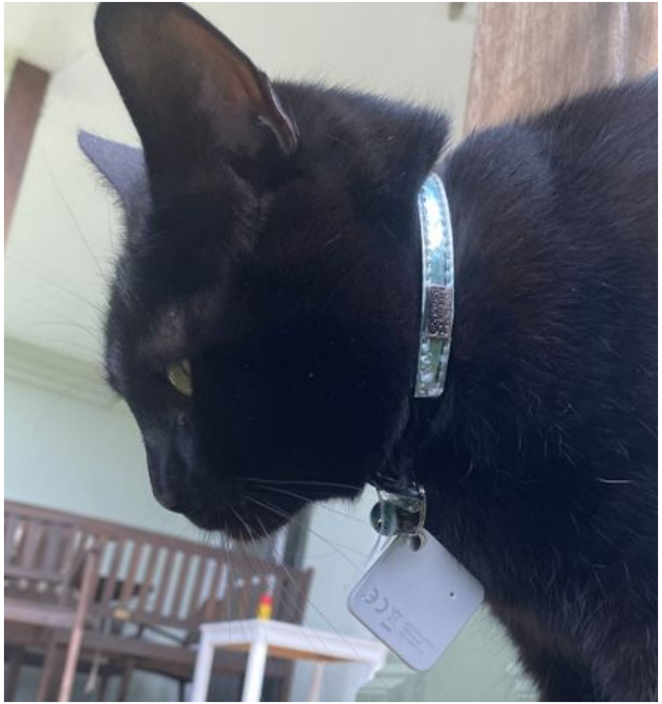
/thoughtworks

@Simon Aubury

What I've learnt & what’s next

- Intuitively think I’m missing features
  - Rain on the ground and WFH status
  - Humidity is not the same as raining
  - Are cats actually predictable?

- Can ML predict where my cat is now?
  - Yes!

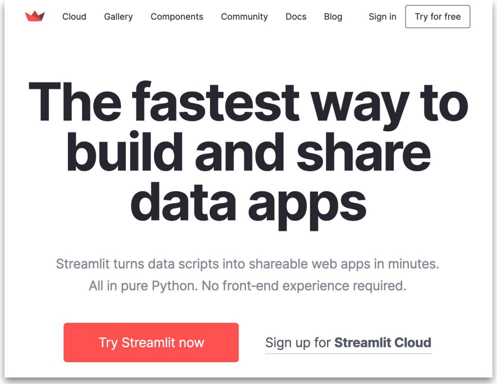


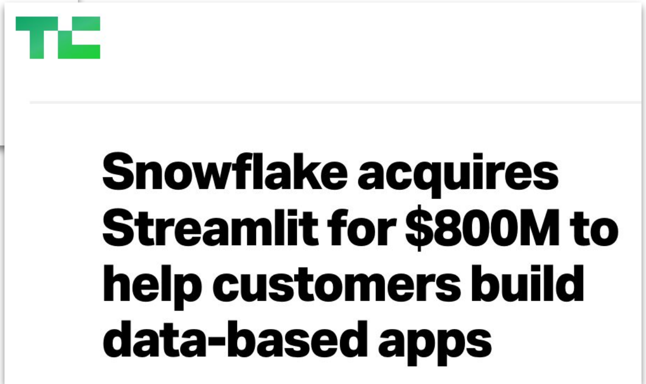
Q & A

https://github.com/saubury/cat-predict

Streamlit

```python
import streamlit as st

# Sidebar
st.set_page_config(initial_sidebar_state='expanded',)

IndoorTemp = st.sidebar.slider('Indoor Temp C', 16 , 28, 22)
OutdoorTemp = st.sidebar.slider('Outdoor Temp C', 10 , 30, 22)

@st.cache(allow_output_mutation=True)
def prepClassification():
    ret = pickle.load(open('./cat_predictor_app.pkl', 'rb'))
    print("Classification loaded")
    return ret

# Reads in saved classification model
load_clf = prepClassification()
pdf = pd.DataFrame(data)

# Apply model to make predictions
prediction = load_clf.predict(pdf)
prediction_proba = load_clf.predict_proba(pdf)
predictionText = load_clf.predict(pdf)[0]
```





Streamlit - open-source python library for creating and sharing web apps

Decision Trees

Machine learning are algorithms that learn from examples. I wanted to build a ML model to predict where my cat Snowy was likely to go knowing the temperature and time. You can use this website to predict where she is likely to be by moving the sliders around on the left.

This website uses classification - a predictive model that assigns a class label to inputs, based on many examples it has been trained on from thousands of past observations of time of day, temperature and location.

Summarising data

DBT stuff

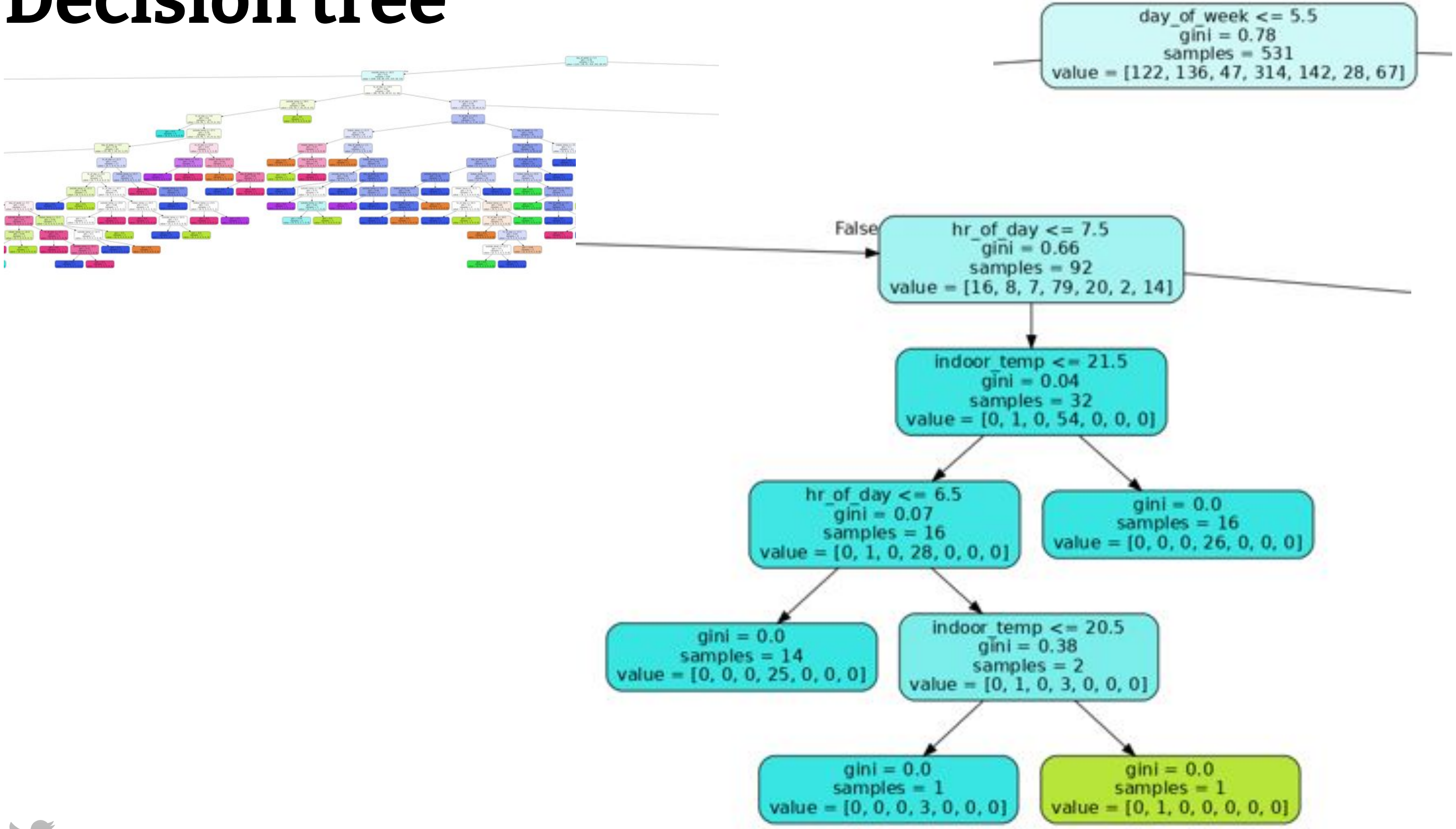
-

```
with cte AS
(
        select created_local_hr
        , state
        , count(*)
        , ROW_NUMBER() OVER (PARTITION BY created_local_hr ORDER BY COUNT(*) DESC) rn
        from state_clean
        where entity_id = 'sensor.snowy_tile'
        group by created_local_hr, state
)
--
select created_local_hr, state as cat_location
```

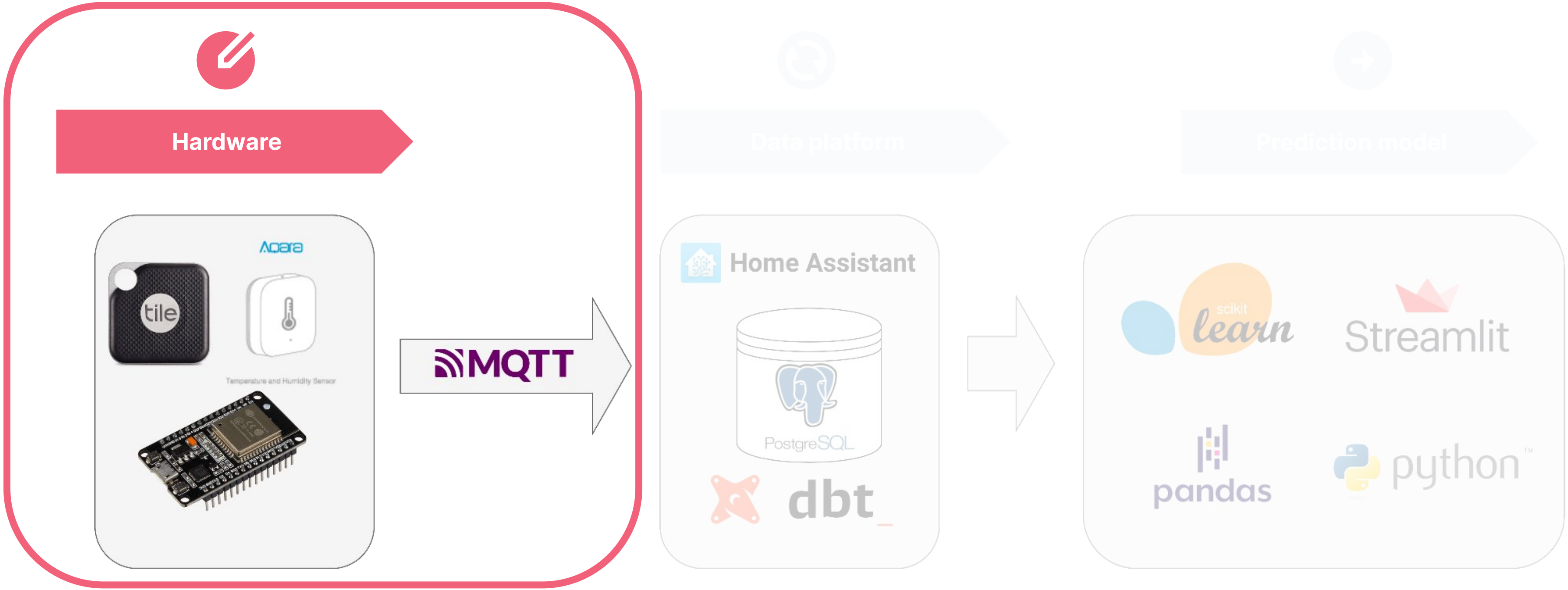
```
03:49:54
03:49:54  1 of 4 START table model hass_schema.state_clean.............................. [RUN]
03:49:54  1 of 4 OK created table model hass_schema.state_clean......................... [SELECT 119681 in 0.20s]
03:49:54  2 of 4 START view model hass_schema.cat_events................................ [RUN]
03:49:54  2 of 4 OK created view model hass_schema.cat_events........................... [CREATE VIEW in 0.05s]
03:49:54  3 of 4 START view model hass_schema.house_events.............................. [RUN]
03:49:54  3 of 4 OK created view model hass_schema.house_events......................... [CREATE VIEW in 0.04s]
03:49:54  4 of 4 START view model hass_schema.all_events................................ [RUN]
03:49:54  4 of 4 OK created view model hass_schema.all_events........................... [CREATE VIEW in 0.04s]
03:49:54
03:49:54  Finished running 1 table model, 3 view models in 0.47s.
```

Data extract

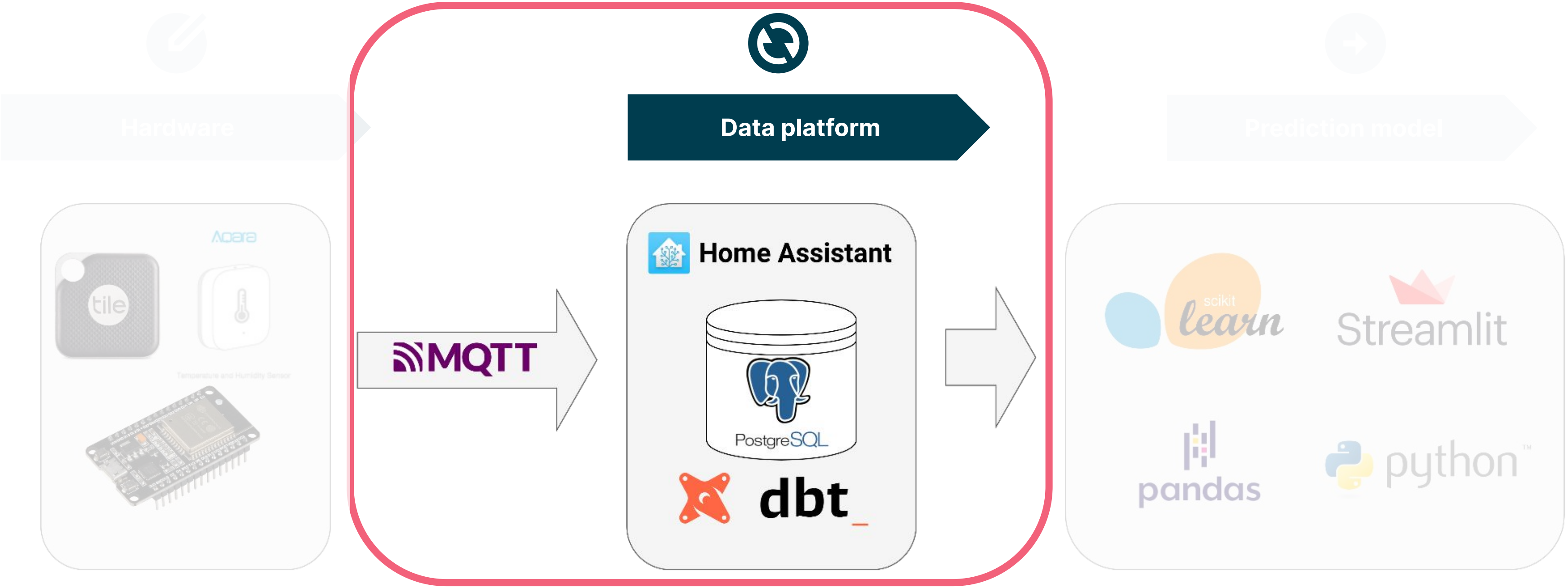
Decision tree



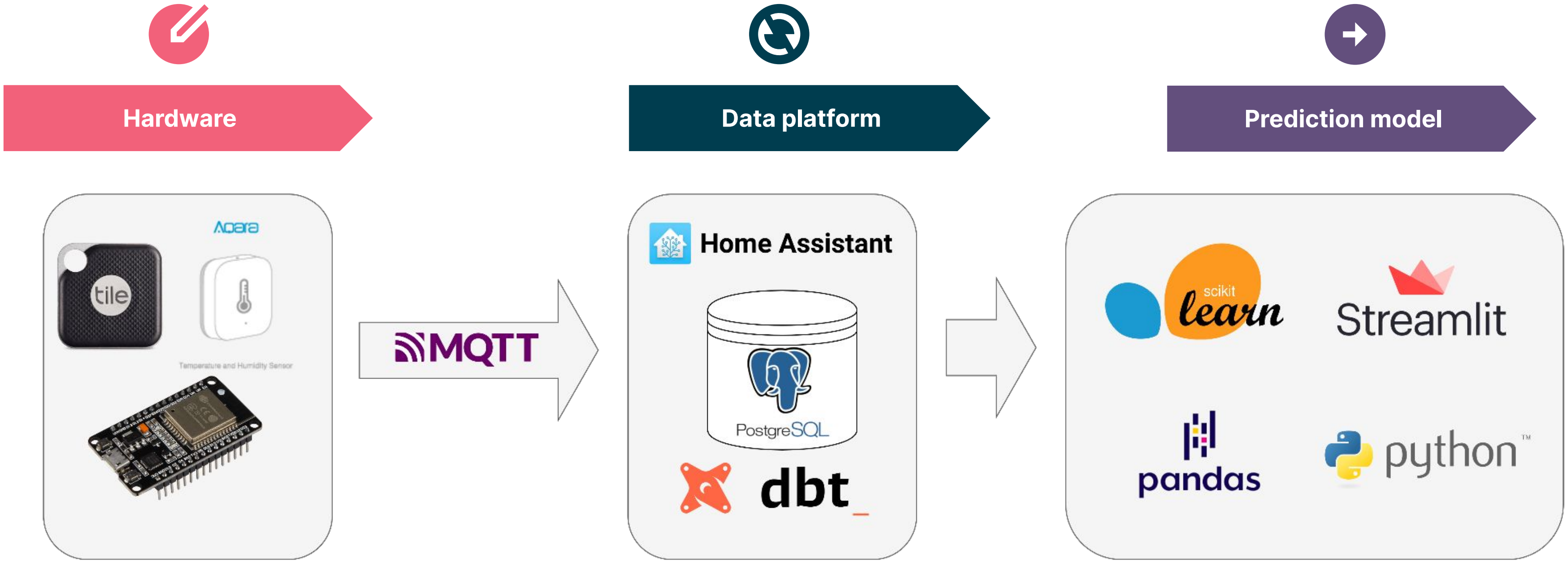
Collect data



Data platform

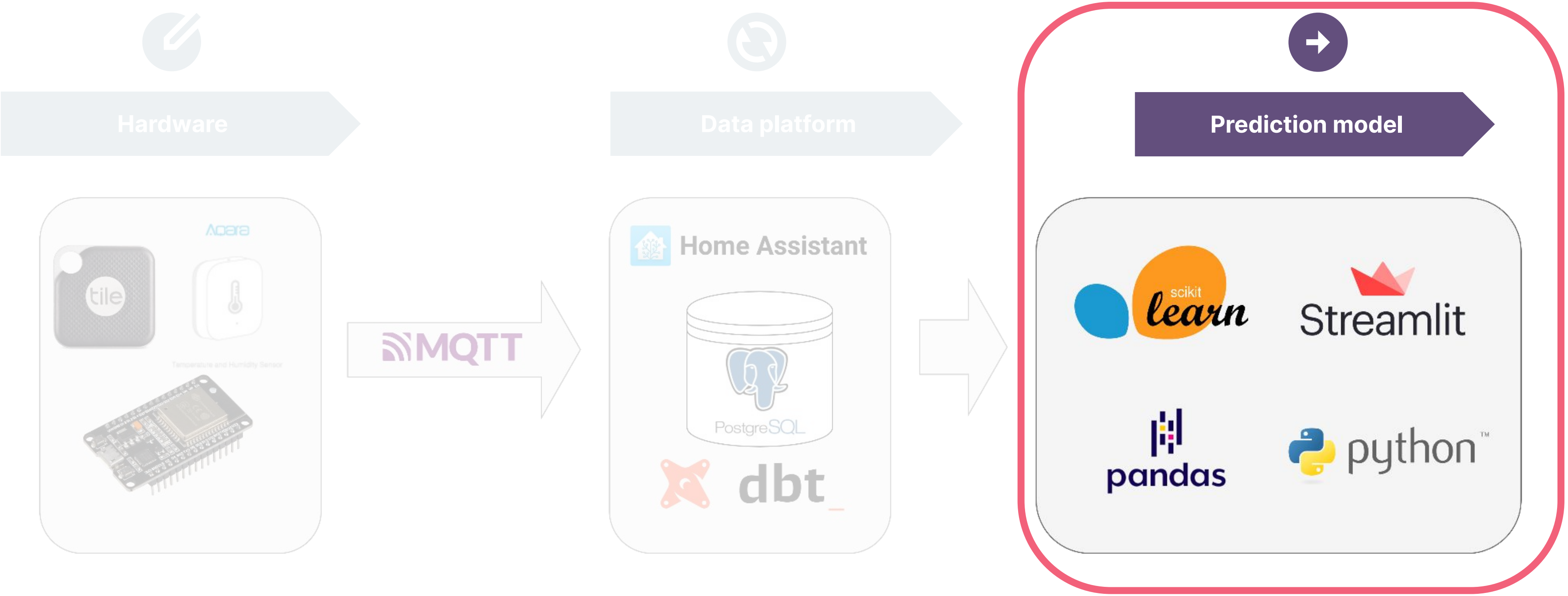


Overview



Hardware
Data platform
Prediction model
Aqara
tile
Temperature and Humidity Sensor
MQTT
Home Assistant
PostgreSQL
dbt
scikit
learn
Streamlit
pandas
python

Prediction model



How to approach most ML problems in 7 steps

Define problem

Obtain, load and explore data
- Exploratory data analysis
- Understand attributes
- Make data visualisations

Prepare and clean data for modelling
- Treat missing values
- Transform data into X and y format
- Split data into train and test set
- Feature engineering

Train model
- Select an algorithm that works for your problem/data
- Use `.fit()`

Evaluate model
- Evaluate accuracy of model
- Use `.score()`, `.confusion_matrix()` and `.classification_report()`

Tune / improve model
- Tune parameters
- Feature engineering
- Use `GridSearchCV()`
- K-folds cross-validation

Iterate!

Use the model
- Interpret coefficients of X variables
- Predict outcomes on new data

Model 1 - Cat in son's room?

Scikit-learn, Logistic Regression

```
# extract just the hour
df['hr_of_day'] = df['created_local_hr'].str[11:13].astype(int)
df['day_of_week'] = pd.to_datetime(df['created_local_hr']).dt.dayofweek.astype(int)
df.drop('created_local_hr', axis=1, inplace=True)
df.dropna(axis=0, how='any', thresh=None, subset=None, inplace=True)

df['is_raining'] = False
# DataFrame.where replace values where the condition is *False*. Read this as "when outside_humidity < 70 is false" then set raining to True
df['is_raining'].where(df['outside_humidity'] < 90.0, True, inplace=True)

df['is_nicho'] = 1
df['is_nicho'].where(df['cat_location'] == 'nicholas', 0, inplace=True)
df.drop('cat_location', axis=1, inplace=True)



display(df.head(5))
#print (df.dtypes)
```

	indoor_temp	outside_temp	outside_humidity	hr_of_day	day_of_week	is_raining	is_nicho
0	24.0	18.0	44.0	4	5	False	0
1	24.0	18.0	45.0	5	5	False	0
2	24.0	17.0	48.0	6	5	False	0
3	23.0	18.0	54.0	7	5	False	0
4	23.0	20.0	53.0	8	5	False	0

```
Accuracy: 0.8844765342960289
Precision: 0.8701298701298701
Recall: 0.7528089887640449
```

Part 4 - Prediction model

/thoughtworks

@Simon Aubury

Part 3 - Data platform

@Simon Aubury

Bootcamp

What are the sort of scenarios we see ML useful for?

Predicting a value/category

Predicting a numerical value based on the sequence of prior values

- How much will this house sell for?
- How long until a specific component in a factory fails?
- Is a user { New, Established, Fickle } ?

Recommendations

Understand the past relationship of users and items so we can suggest new items to users.

- Netflix
- Amazon
- People you might know; LinkedIn

Natural language understanding

NLU is an obvious place to build on for clients that already have lots of text

- Very close relationship to search

Image understanding

Image understanding has made huge progress in last 5 years

- Convolutional neural networks are the key technique
- Provide the ability to convert from images to numerical representation