SAOUG CONFERENCE 2025

ADD LOGO

HELLO
AFRICA

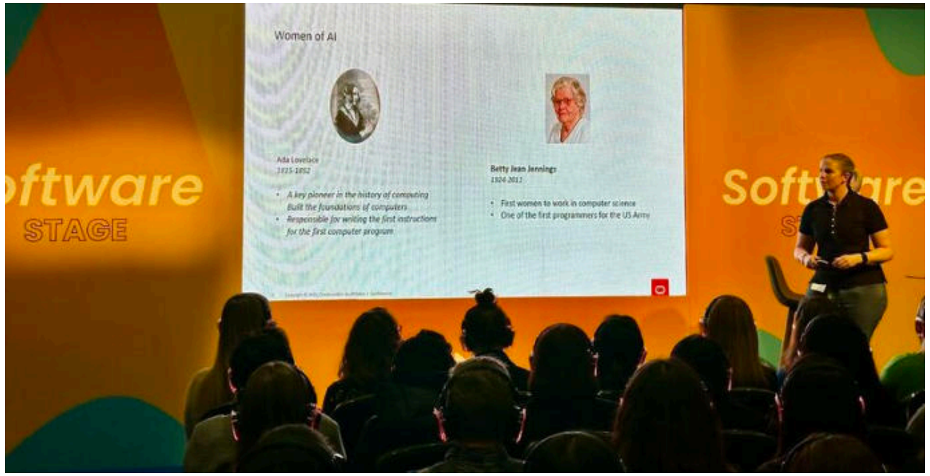
Game Changers Python Video Analysis

Abi Giles-Haigh

Innovation and Analytics Director

Add logo

- PhD in ML – Computation Modelling of the human heart
- Oracle DBA for 5 years
- Worked public health care, private banking



Oracle ACE

Tech Superstars Unite

Get worldwide recognition as an Oracle ACE


Oracle.com profile page


Swag, certification exam credit & event passes


Exclusive content


Networking events


Your own Oracle cloud account


Travel support

@oracleace

Linkedin.com/groups/72183

@oracleace.bsky.social

Learn more at:
ace.oracle.com

Add logo

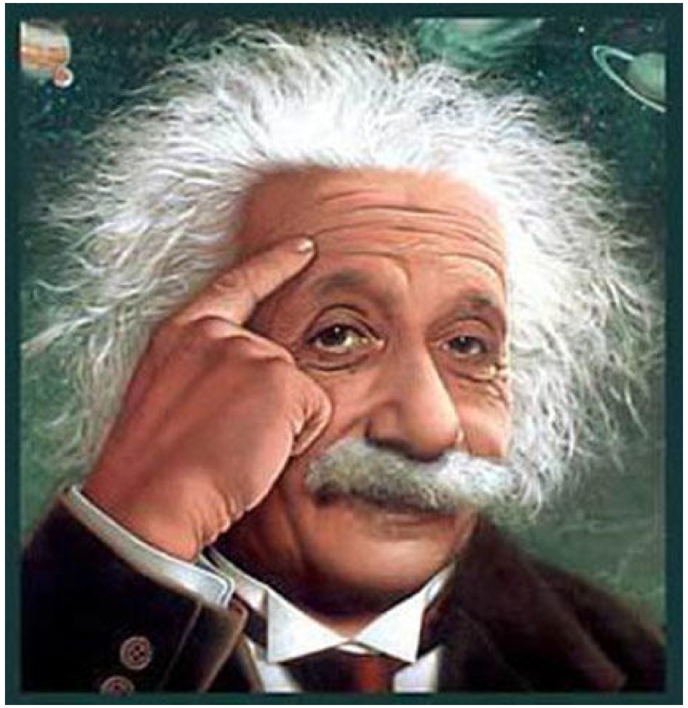
Add logo

Business problem

Start with a Well-Defined Business Problem Statement

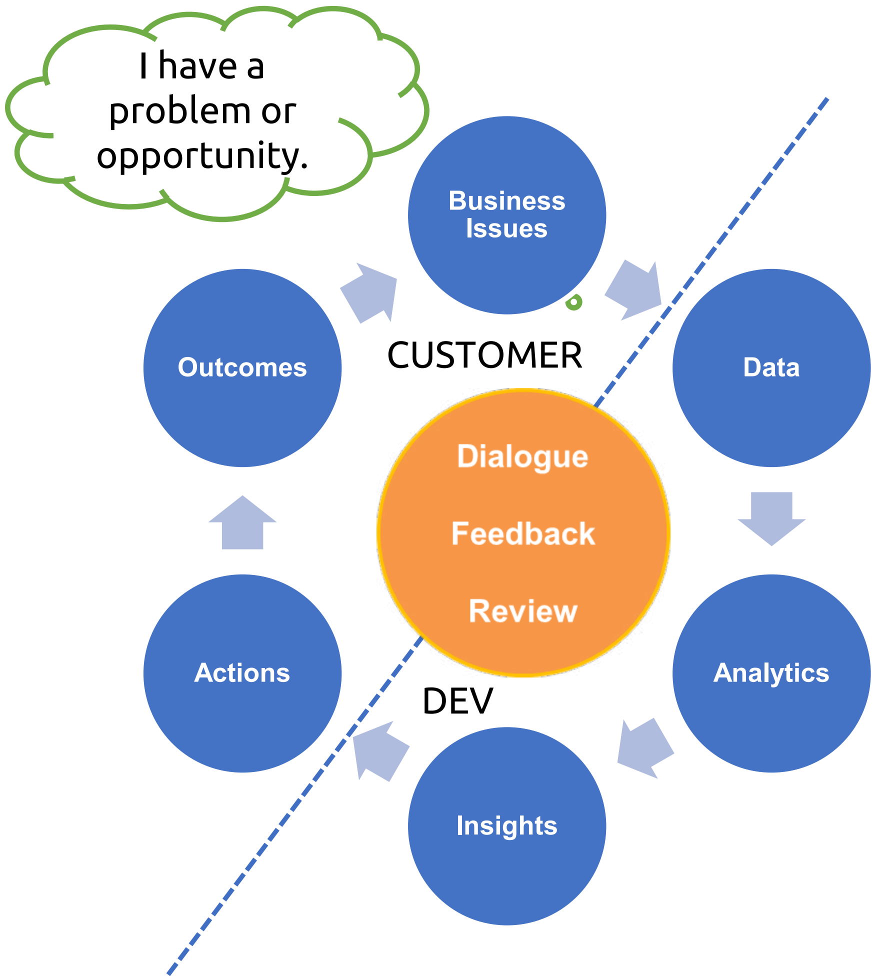
“If I had an hour to solve a problem, I'd spend 55 minutes thinking about the problem and 5 minutes thinking about solutions.”

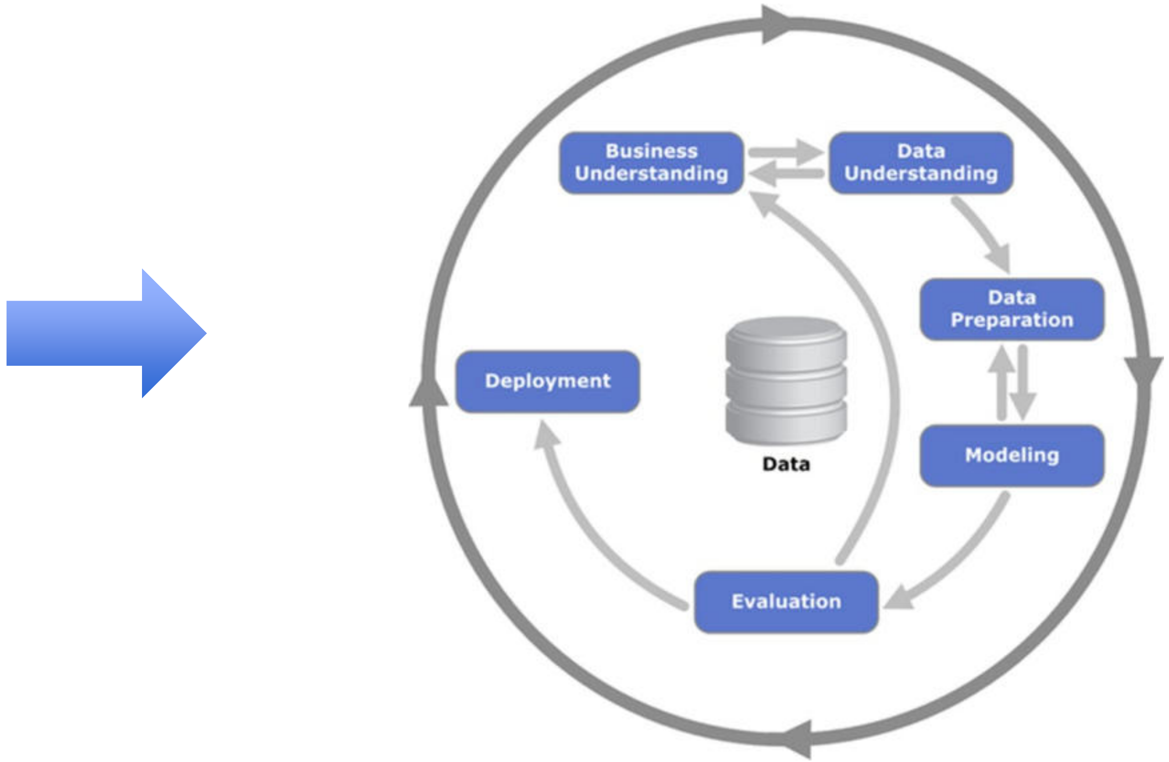
— Albert Einstein



Add logo

Business need and operational process



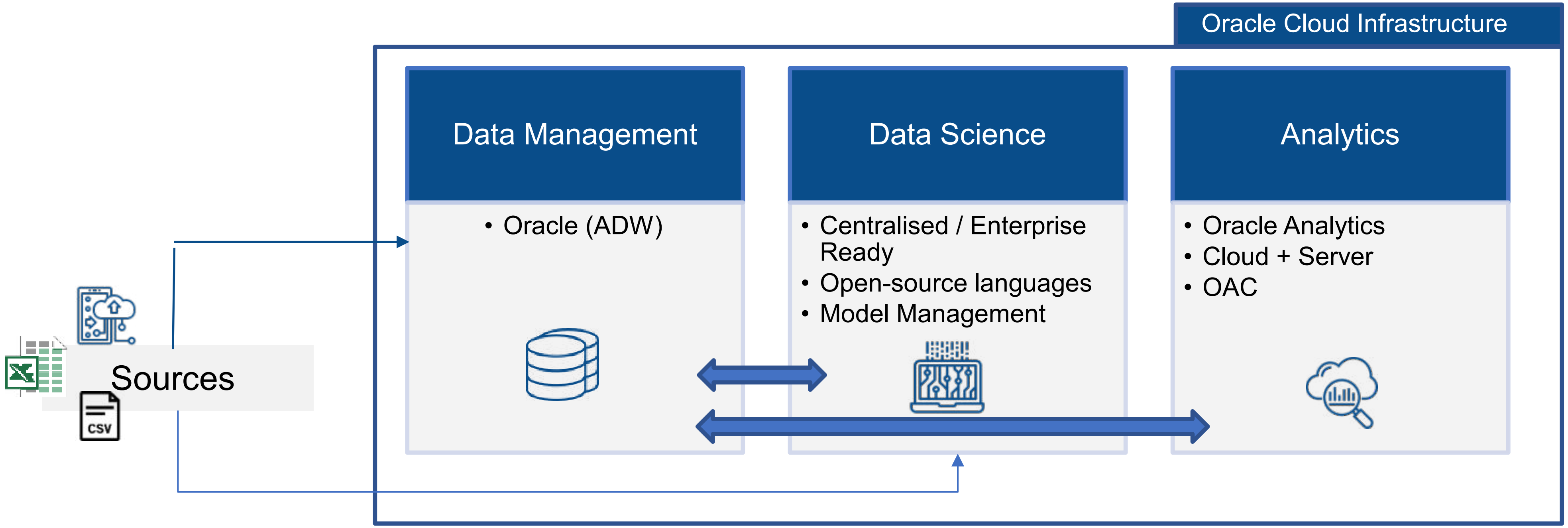


- The operating model must be flexible to respond to the needs of the customers

- Cross-industry standard process for data mining (CRISP-DM)
- Data mining is a process of extracting and discovering patterns in large data sets

Add logo

Architecture



Oracle OCI: VM.Standard.E2.2 – Data Science

Clone Oracle Database Environment

- Database connection – pull and push statistics from the videos

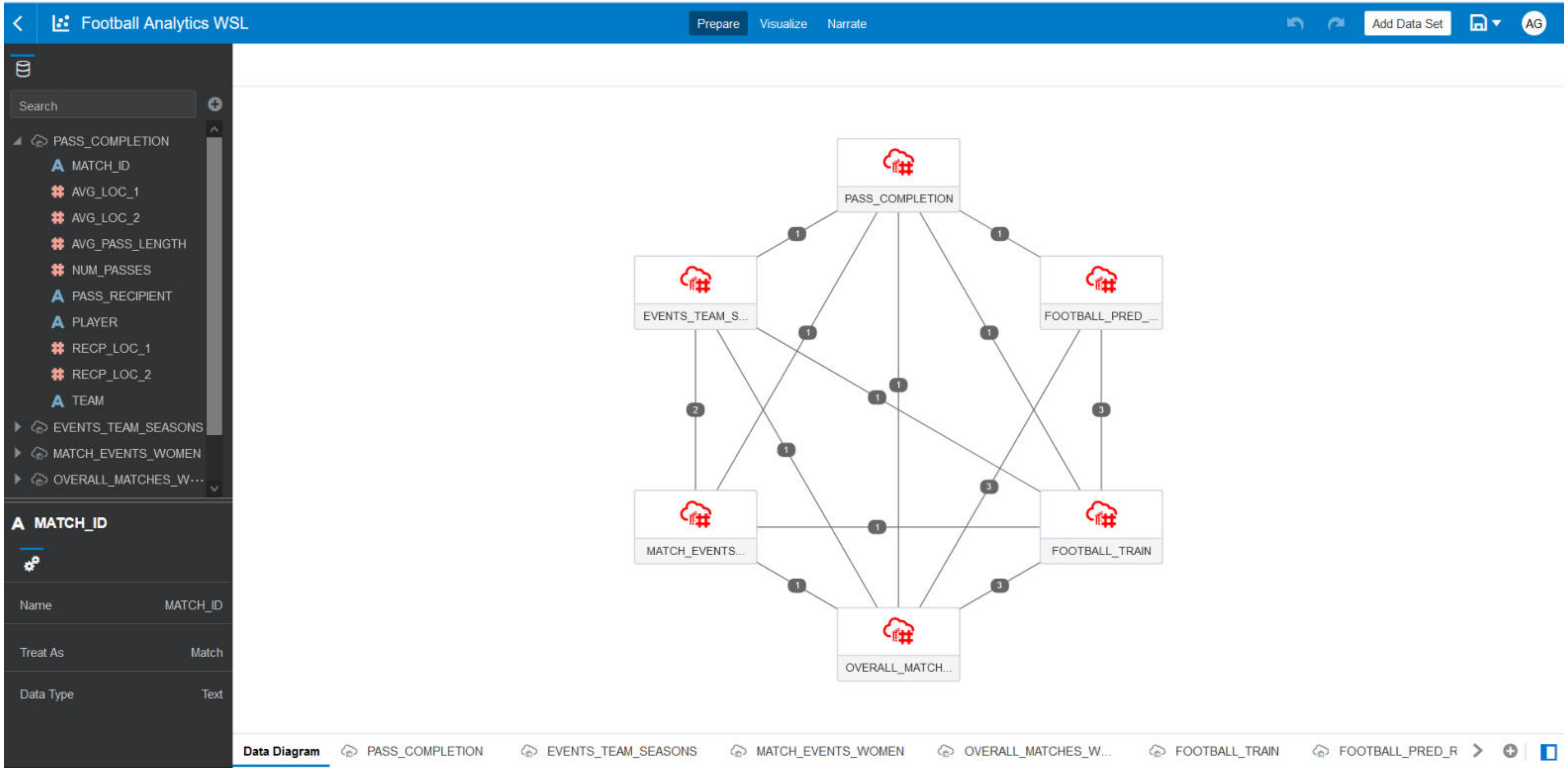
Install Python Packages – Pip Install….

- sudo yum install freeglut-devel *#OpenGL Utility Toolkit*
- pip install opencv-python *#Read the videos*
- pip install matplotlib *#Visulaisation tools*
- pip install pandas *#Data Processing*

Add logo

Descriptive analytics

Data Diagrams



Add logo

Descriptive – Passing Networks

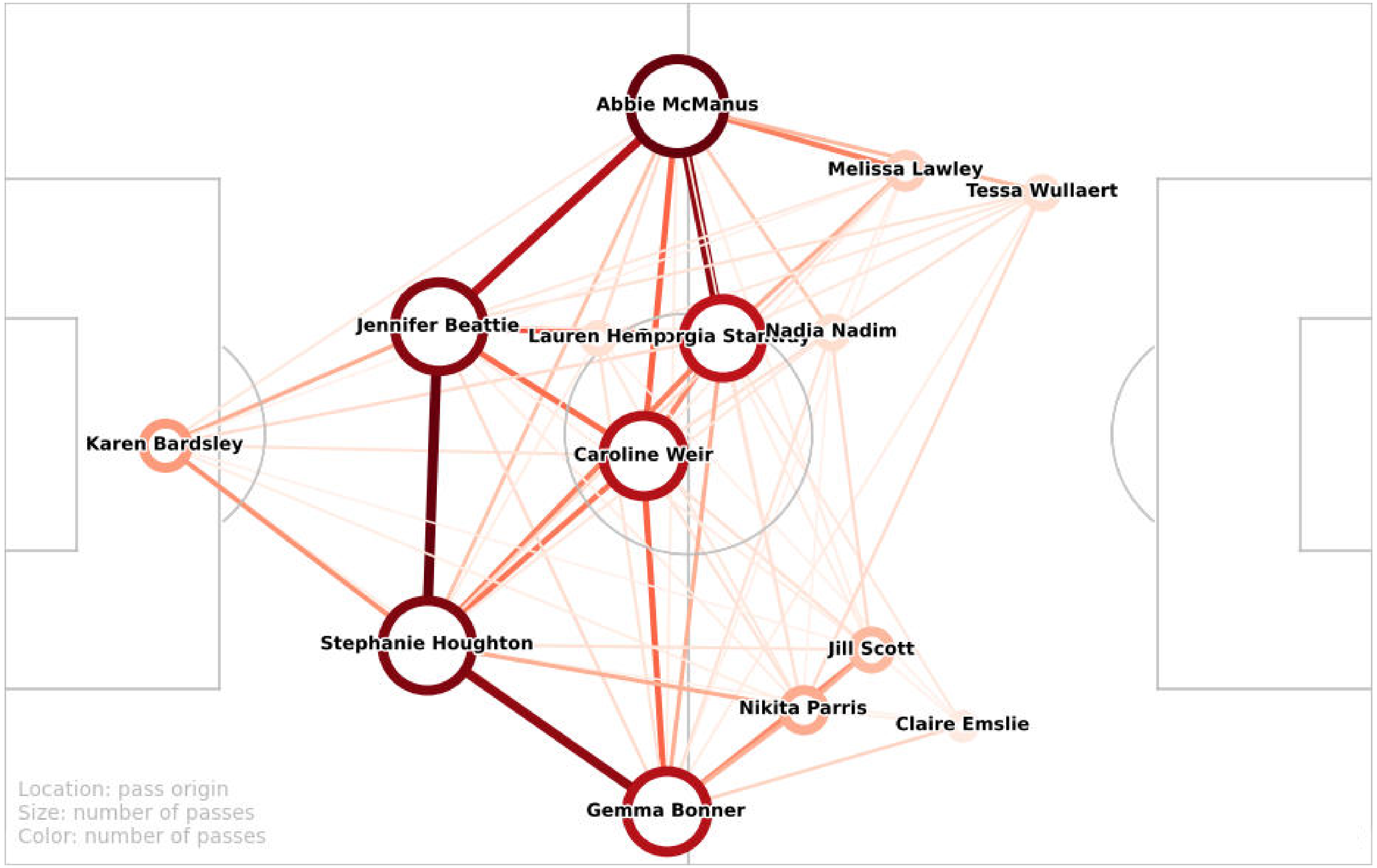
Number of passes completed from player A to player B

Average location on the pitch

Highlights where they were during the game as well as potential threats to deal with

Mapped to a 2D pitch to help players understand their location

Manchester City WFC's passing network against Everton LFC



Enter OCI & Python....

```
[1]: # Make sure that the path is the root of the project
     %cd ..

     /home/datascience/Football Analytics

[2]: from pandas.io.json import json_normalize
     from utils import read_json

[3]: import cx_Oracle
     import logging
     import os
     import pandas as pd
     import warnings

     from ads.database import connection
     from ads.database.connection import Connector
     from ads.dataset.factory import DatasetFactory
     from ads.vault.vault import Vault
     import sqlalchemy
     from sqlalchemy import create_engine
     from sqlalchemy.ext.declarative import declarative_base
     from sqlalchemy import Column, Integer, String, Numeric
     from sqlalchemy.orm import sessionmaker
     from urllib.request import urlopen

     warnings.filterwarnings("ignore", category=DeprecationWarning)
     logging.basicConfig(format='%(levelname)s:%(message)s', level=logging.INFO)

[2]: # Sample credentials that are going to be stored.
     credential = {'database_name': 'db202004201349_high',
```

Step 2: Set text information

```
[13]: plot_name = "statsbomb_match{0}_{1}".format(match_id, team_name)

      opponent_team = [x for x in df_events.team_name.unique() if x != team_name][0]
      plot_title ="{0}'s passing network against {1} ".format(team_name, opponent_team)

      plot_legend = "Location: pass origin\nSize: number of passes\nColor: number of passes"
```

Step 3: Prepare data

```
[14]: def _statsbomb_to_point(location, max_width=120, max_height=80):
          '''
          Convert a point's coordinates from a StatsBomb's range to 0-1 range.
          '''
          return location[0] / max_width, 1-(location[1] / max_height)

[15]: df_passes = df_events[(df_events.type_name == "Pass") &
                            (df_events.pass_outcome_name.isna()) &
                            (df_events.team_name == team_name)].copy()

      # If available, use player's nickname instead of full name to optimize space in plot
      df_passes["pass_recipient_name"] = df_passes.pass_recipient_name.apply(lambda x: names_dict[x] if names_dict[x] else x)
      df_passes["player_name"] = df_passes.player_name.apply(lambda x: names_dict[x] if names_dict[x] else x)

      df_passes.head()
```

Enter OCI & Python....

```
[16]: df_passes["origin_pos_x"] = df_passes.location.apply(lambda x: _statsbomb_to_point(x)[0])
df_passes["origin_pos_y"] = df_passes.location.apply(lambda x: _statsbomb_to_point(x)[1])
player_position = df_passes.groupby("player_name").agg({"origin_pos_x": "median", "origin_pos_y": "median"})

player_position
```

[16]:

player_name	origin_pos_x	origin_pos_y
Abbie McManus	0.491667	0.88125
Caroline Weir	0.466667	0.47500
Claire Emslie	0.700000	0.16250
Gemma Bonner	0.483333	0.06250
Georgia Stanway	0.525000	0.61250
Jennifer Beattie	0.316667	0.62500
Jill Scott	0.633333	0.25000
Karen Bardsley	0.116667	0.48750
Lauren Hemp	0.433333	0.61250
Melissa Lawley	0.658333	0.80625
Nadia Nadim	0.604167	0.61875
Nikita Parris	0.583333	0.18125
Stephanie Houghton	0.308333	0.25625

```
[18]: df_passes["pair_key"] = df_passes.apply(lambda x: "_".join(sorted([x["player_name"], x["pass_recipient_name"]])), axis=1)
pair_pass_count = df_passes.groupby("pair_key").size().to_frame("num_passes")
pair_pass_value = df_passes.groupby("pair_key").size().to_frame("pass_value")

pair_pass_count.head(10)
```

[18]:

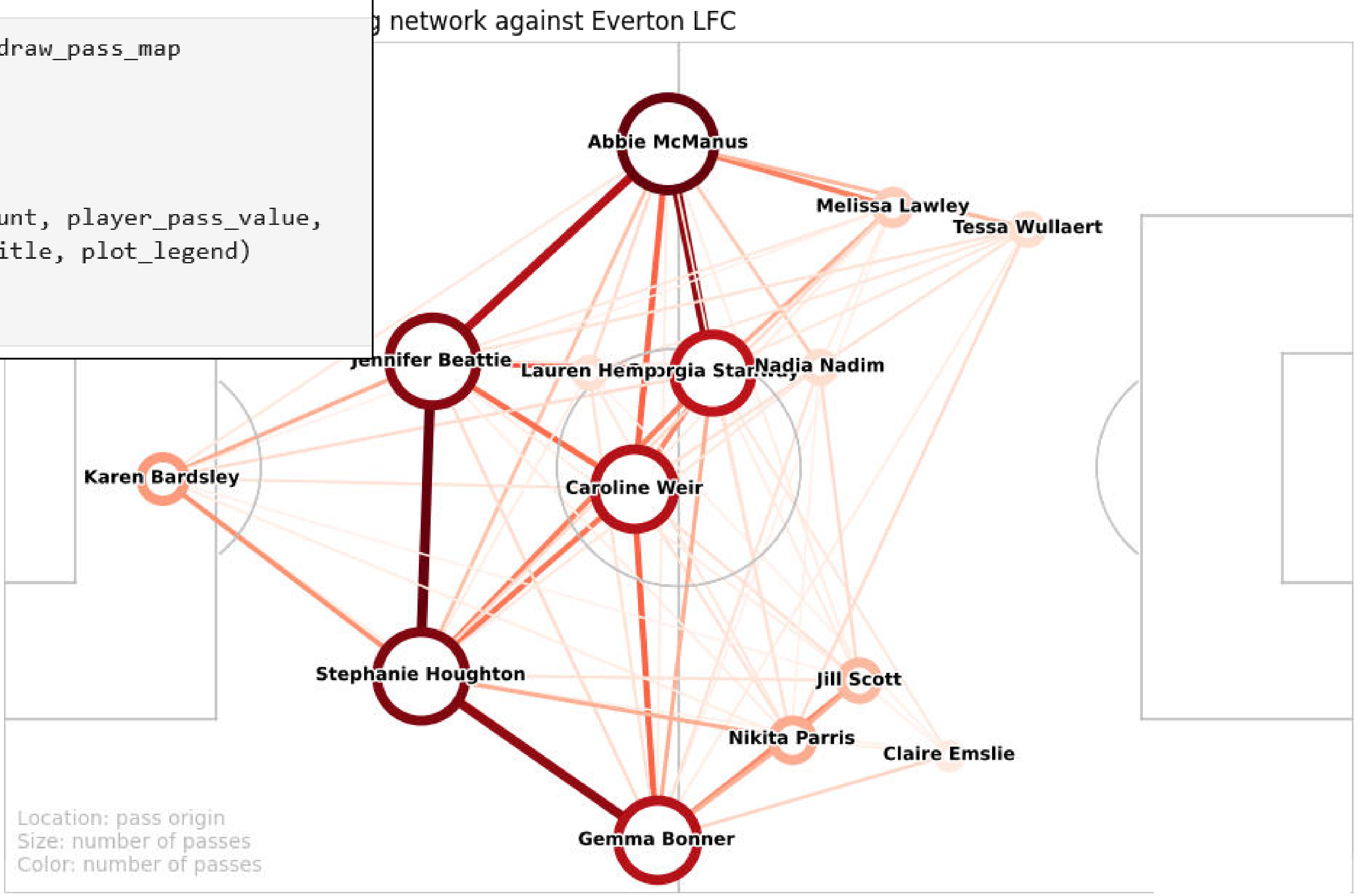
pair_key	num_passes
Abbie McManus_Caroline Weir	18
Abbie McManus_Gemma Bonner	2
Abbie McManus_Georgia Stanway	32
Abbie McManus_Jennifer Beattie	29
Abbie McManus_Jill Scott	2
Abbie McManus_Karen Bardsley	3
Abbie McManus_Lauren Hemp	5
Abbie McManus_Melissa Lawley	15
Abbie McManus_Nadia Nadim	7
Abbie McManus_Nikita Parris	2

Step 4: Plot passing network

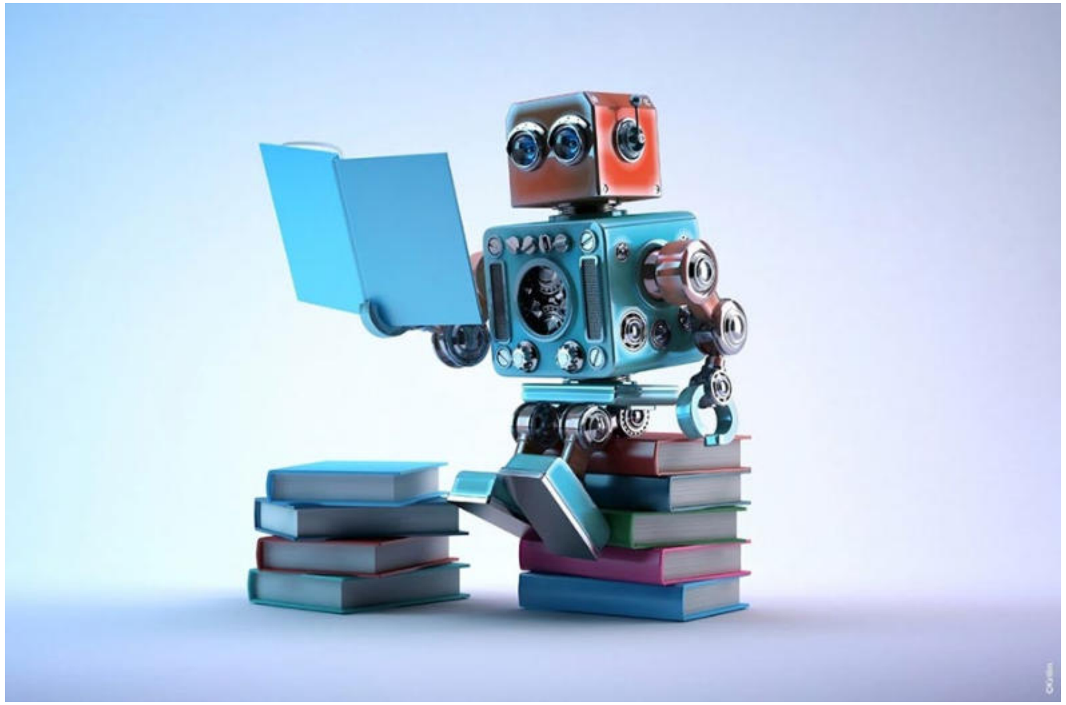
[19]:
```python
from visualization.passing_network import draw_pitch, draw_pass_map
import matplotlib.pyplot as plt


ax = draw_pitch()
ax = draw_pass_map(ax, player_position, player_pass_count, player_pass_value,
              pair_pass_count, pair_pass_value, plot_title, plot_legend)

plt.savefig("demo/{0}.png".format(plot_name))
```



Augment Analytics: Machine Learning



Present historic data to a machine

Identify patterns within the data

Unsupervised:

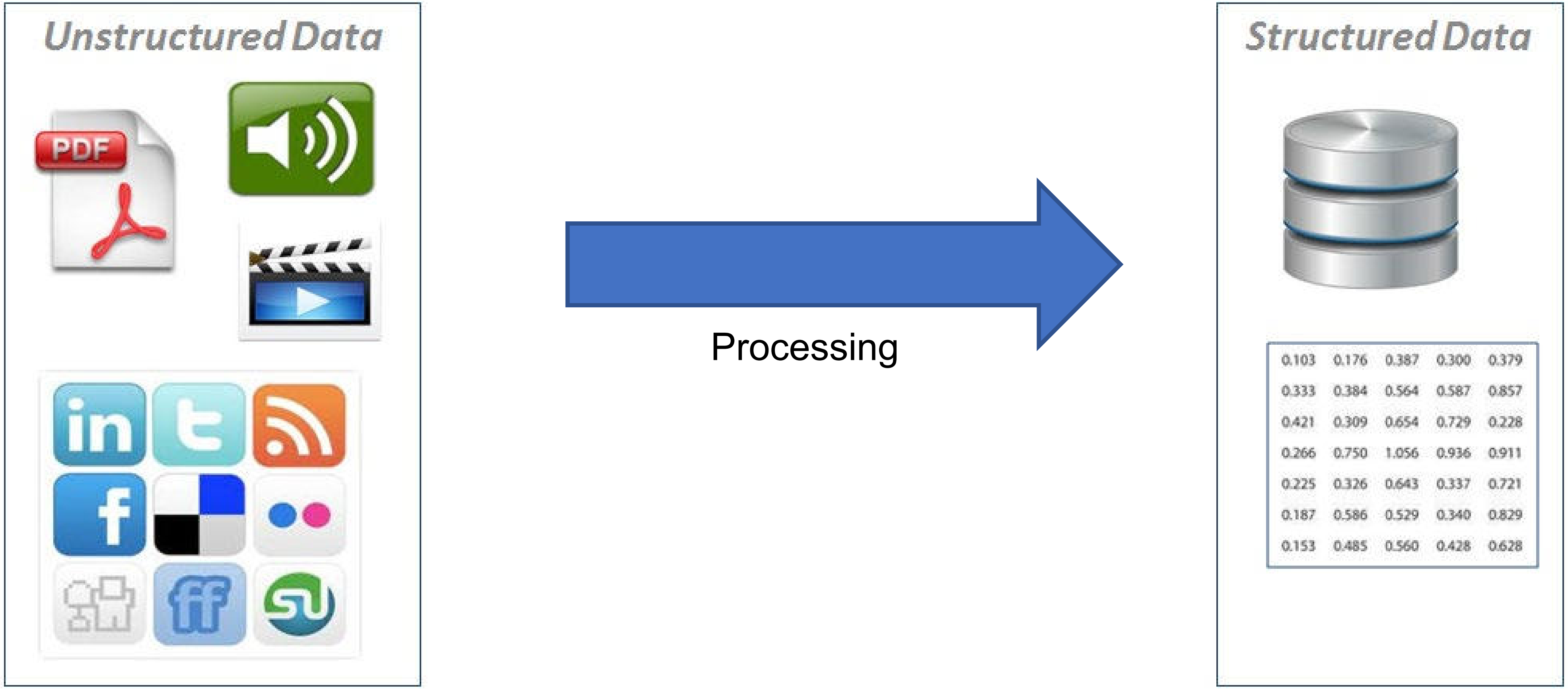
- Identify unusual patterns

Supervised:

- Learning from a particular target

Bad Data + Machine Learning = Really Big Problem!

From Video to Stats.....



0.103	0.176	0.387	0.300	0.379
0.333	0.384	0.564	0.587	0.857
0.421	0.309	0.654	0.729	0.228
0.266	0.750	1.056	0.936	0.911
0.225	0.326	0.643	0.337	0.721
0.187	0.586	0.529	0.340	0.829
0.153	0.485	0.560	0.428	0.628

Add logo

How its traditionally captured

Human labelling of data

Utilising gaming theory – reward/ games consoles

Requires time/ training

Still used by betting companies etc to provide a live feedback from the game

Add logo

How its currently captured

Fixed cameras around the arena/ stadium

Players/ equipment have GPS trackers on/ match with the visual input - cameras

Location vs pitch location enabling basic trigonometry to be used in combination with GPS

Then utilises AI / ML to detect different players and equipment/ areas

Add logo

Deep learning – Image Processing

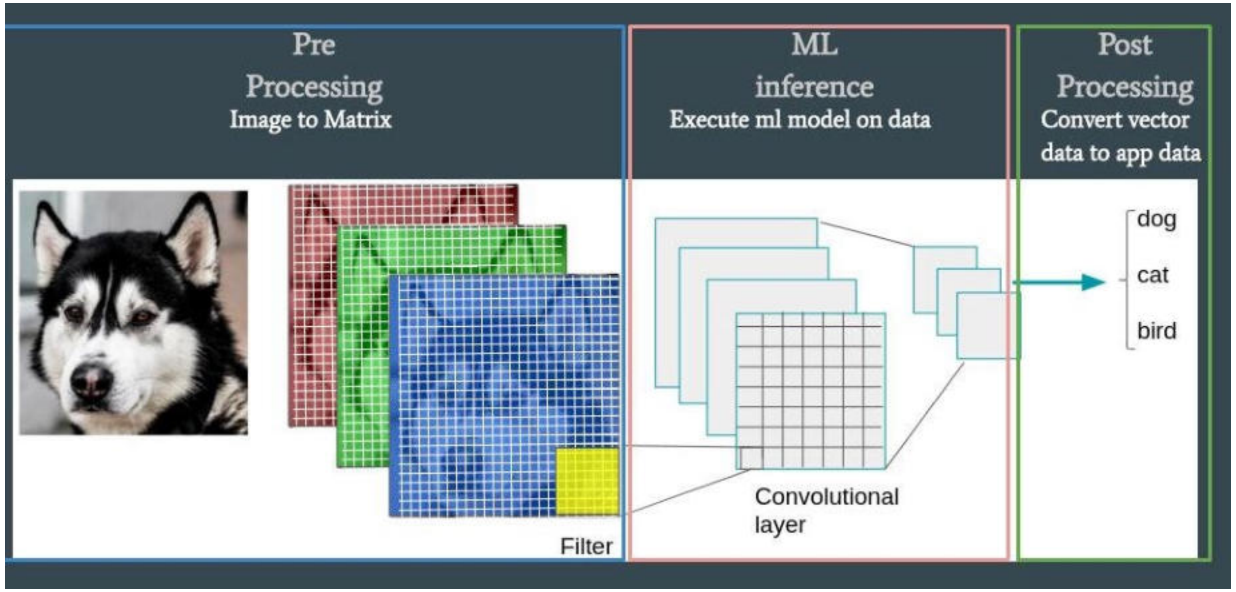
Videos are simply lots of images together

Decide how often to extract frames

Train the ML model

Apply the model to the video

Extract the results



How to detect players

Difference between team 1 and 2

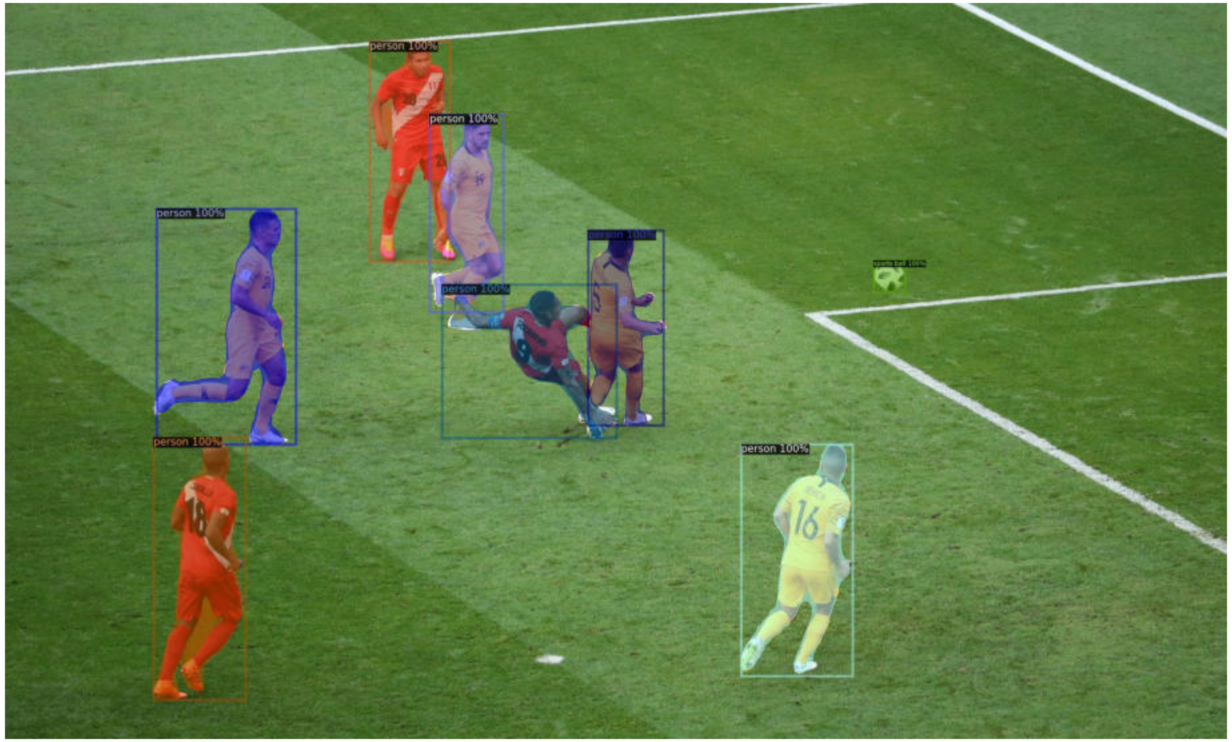
- Red or Blue

Ball

- Colour and size

Pitch

- White lines on the pitch
- Extrapolate to a 2D pitch



Add logo

Original Video: England v USA Women 2019

Add logo

```python
#Import libraries
import cv2
import os
import numpy as np

#Reading the video
vidcap = cv2.VideoCapture('USA_Womens.mp4')
success,image = vidcap.read()
count = 0
success = True
idx = 0
print('we start')
```

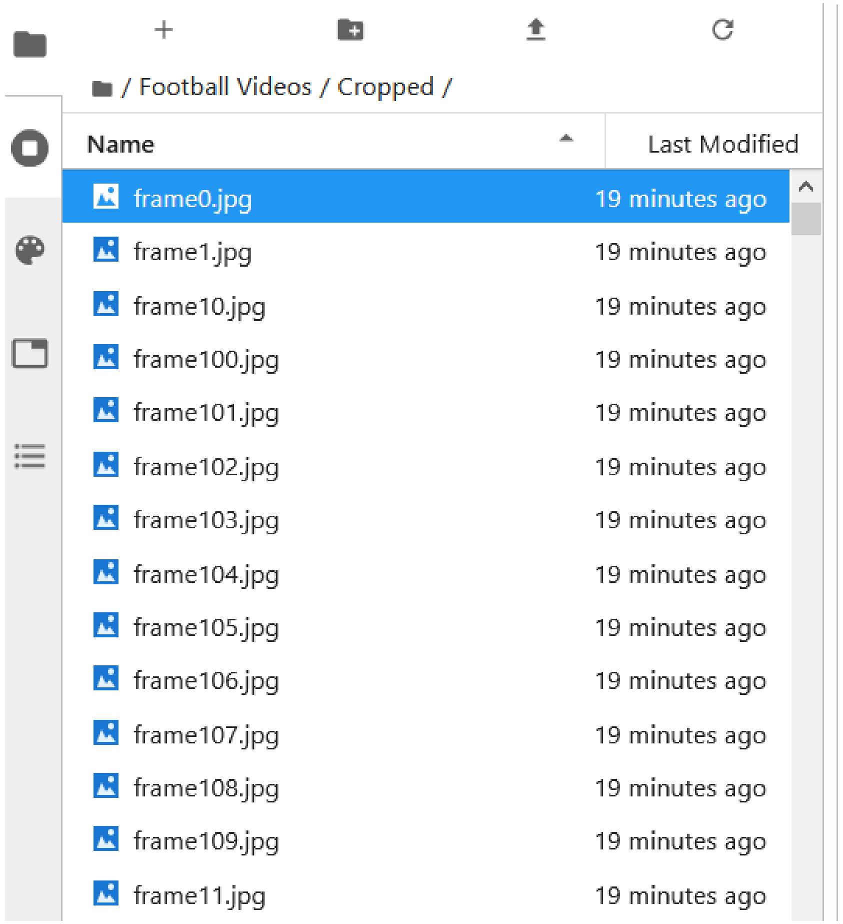
```python
    #Red range
    lower_red = np.array([0,31,255])
    upper_red = np.array([176,255,255])

    #white range
    lower_white = np.array([0,0,0])
    upper_white = np.array([0,0,255])
```

```python
        #Detect players
        if(h>=(1.5)*w):
            if(w>15 and h>= 15):
                idx = idx+1
                player_img = image[y:y+h,x:x+w]
                player_hsv = cv2.cvtColor(player_img,cv2.COLOR_BGR2HSV)
                #If player has white jersy
                mask1 = cv2.inRange(player_hsv, lower_white, upper_white)
                res1 = cv2.bitwise_and(player_img, player_img, mask=mask1)
                res1 = cv2.cvtColor(res1,cv2.COLOR_HSV2BGR)
                res1 = cv2.cvtColor(res1,cv2.COLOR_BGR2GRAY)
                nzCount = cv2.countNonZero(res1)
```

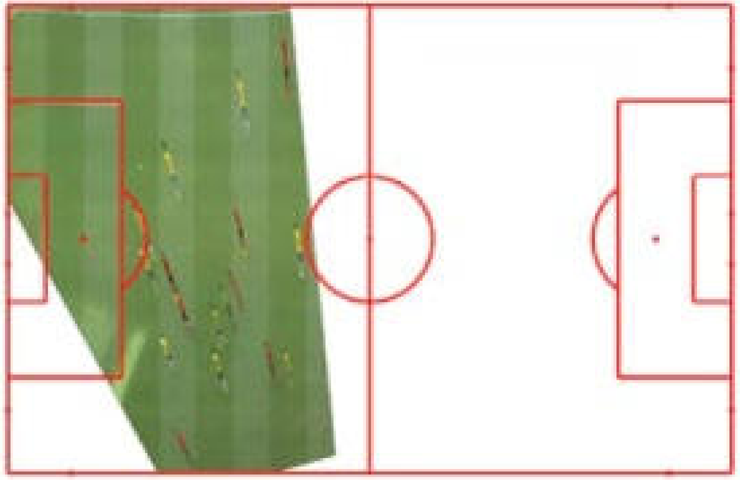
```
    cv2.imwrite("./Cropped/frame%d.jpg" % count, image)
    print('Read a new frame: ', success)     # save frame as JPEG file
    count += 1
    #cv2.startWindowThread()
    #cv2.namedWindow("preview")
    #cv2.imshow('Match Detection',image)
    #if cv2.waitKey(1) & 0xFF == ord('q'):
        #break
    success,image = vidcap.read()

vidcap.release()
cv2.destroyAllWindows()
```



Processed: England v USA Women 2019

Location, Location, Location...



Teach the machine to look for edge of the pitch

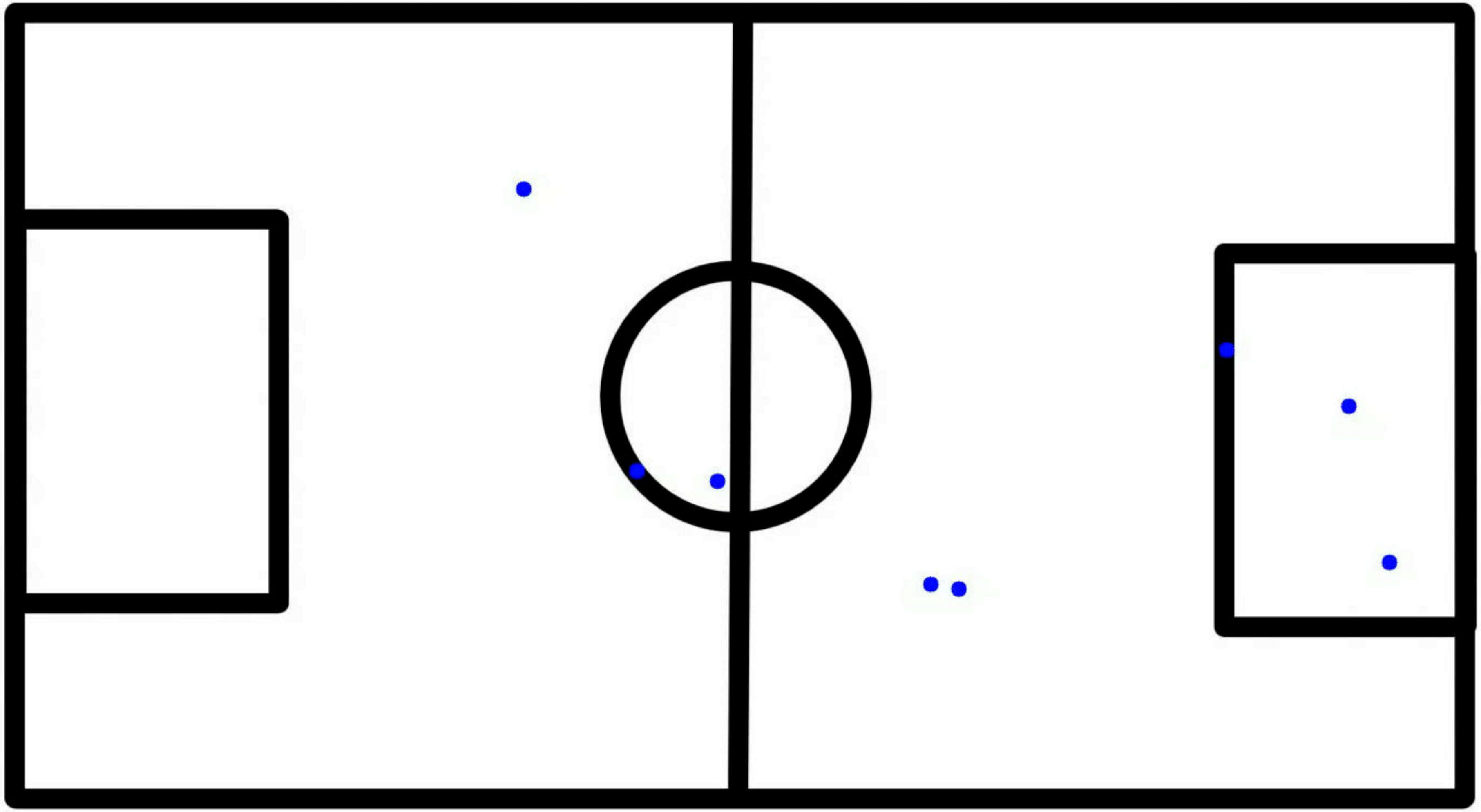
Use the lines detected to map to a 2D pitch

Use some clever maths (Homography matrices) to figure out the location on the pitch

Plot the player each frame of the video

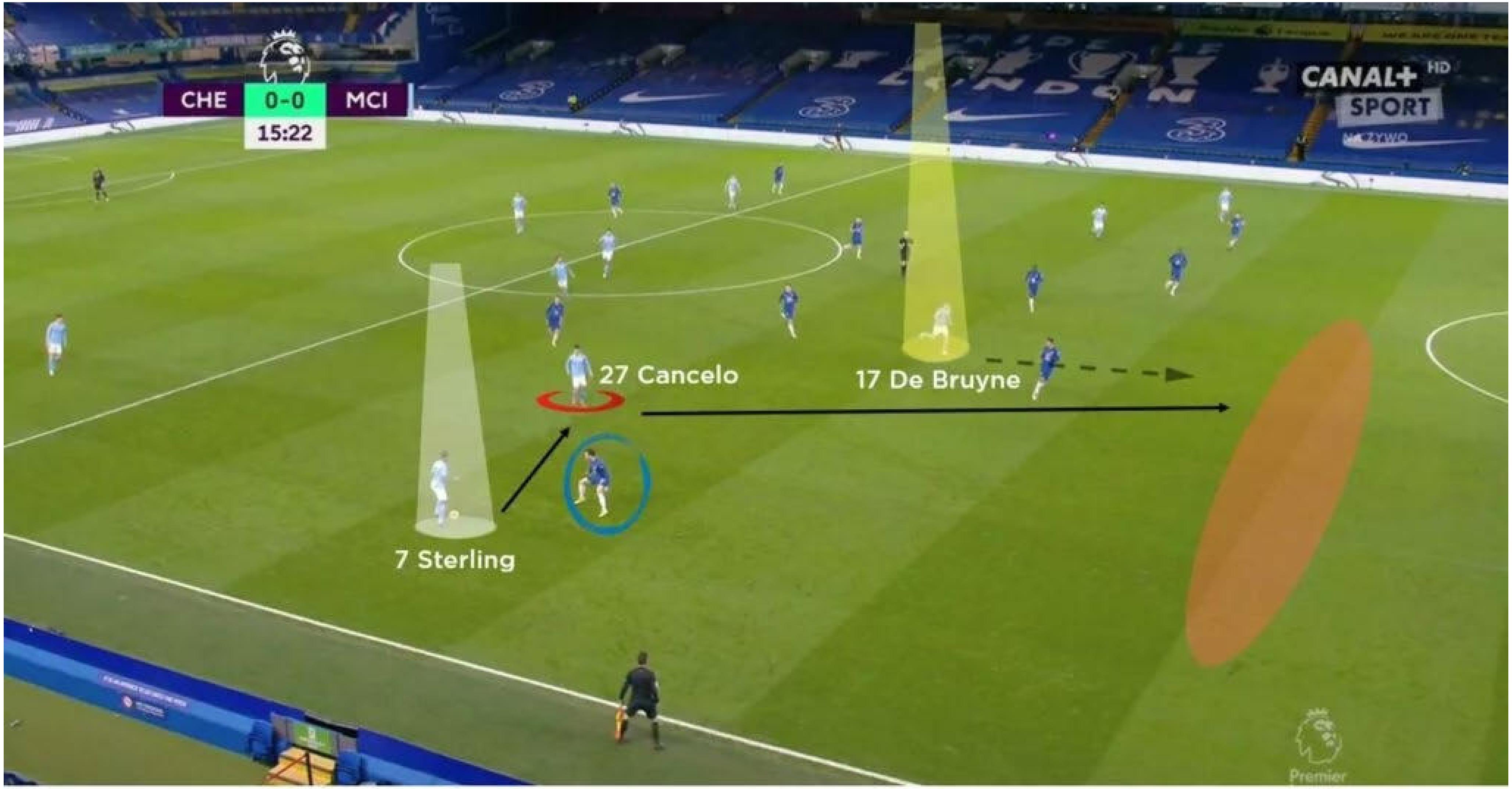
Add logo

Piecing it all together



Add logo

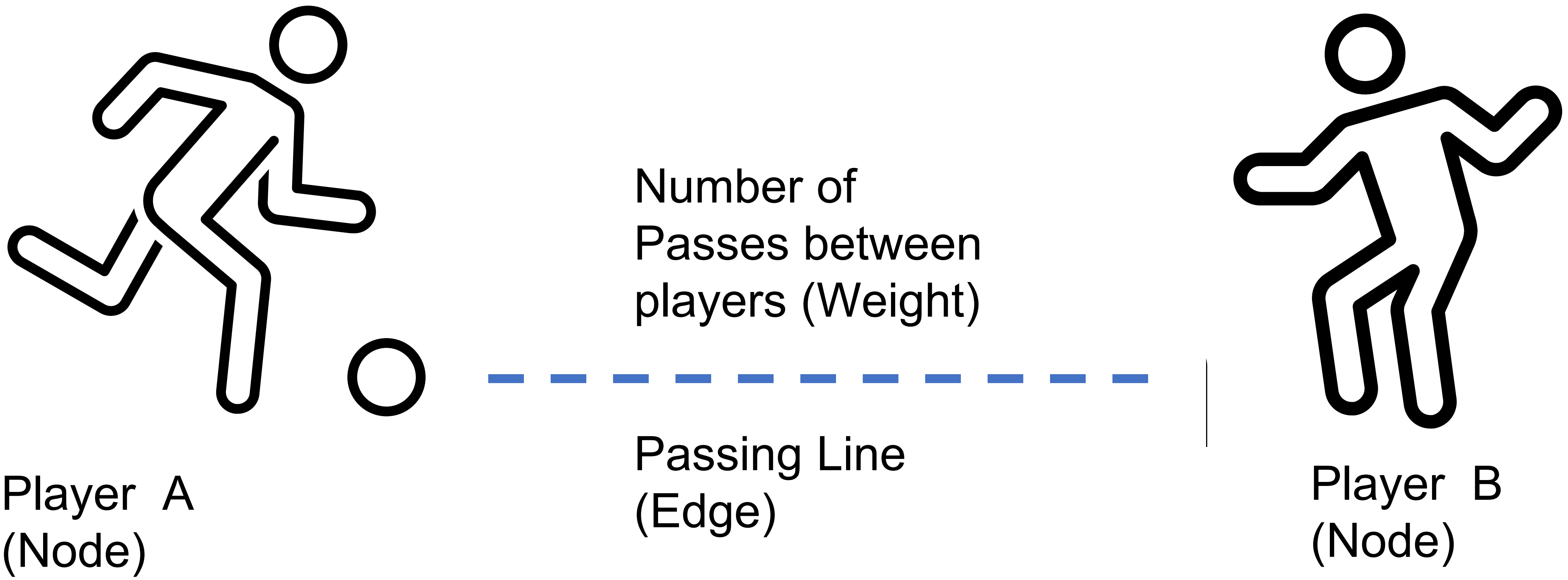
Add logo



Powered by wyscout

Add logo

What does the property graph look like



Put the passes together

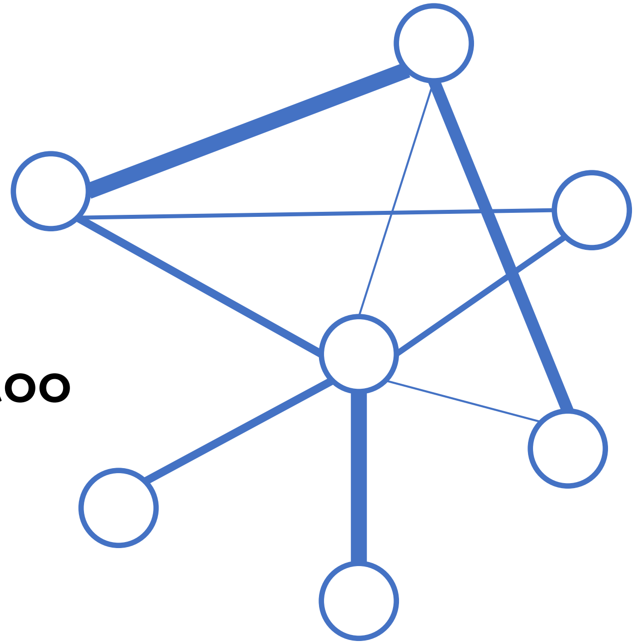
Nodes for each player

Edges for each pass

Weights for the edges:
- **number of passes between players**
- **Distance of the pass**

Degrees for the number of players a player receives or passes too

All combined, it makes a passing graph



Add logo

```
SELECT match_id,player_id, player, team, COUNT(1) AS Num_Passes_Recieved
  FROM graph_table ( WWC23_Passing_Graph
    MATCH (src is WWC23_Players) –
                  [e IS WWC23_Pass_Transactions] -> (dst is WWC23_Players)
    COLUMNS (dst.player_id, dst.player, dst.team, e.match_id)
  ) GROUP BY match_id, player_id, player, team
ORDER BY Num_Passes_Recieved DESC
FETCH FIRST 10 ROWS ONLY;
```

MATCH_ID	PLAYER_ID	PLAYER	TEAM	NUM_PASSES_RECIEVED
3893799	10261	Sara Doorsoun-Khajeh	Germany Women's	113
3893799	10399	Kathrin Julia Hendrich	Germany Women's	103
3893799	32355	Felicitas Rauch	Germany Women's	90
3901797	4999	Lindsey Michelle Horan	United States Women's	88
3904628	10161	María Francesca Caldentey Oliver	Spain Women's	85
3893828	4642	Millie Bright	England Women's	84
3893828	10252	Alex Greenwood	England Women's	82
3901734	25638	Moeka Minami	Japan Women's	81
3893820	10395	Maren Nævdal Mjelde	Norway Women's	81
3893833	10399	Kathrin Julia Hendrich	Germany Women's	79

Add logo

You can find players involved in a chain of passes

```
SELECT player_id, player, team, COUNT(1) AS Num_In_Middle
  FROM graph_table (WWC23_Passing_Graph
    MATCH (src is WWC23_Players) - [e IS WWC23_Pass_Transactions] ->
          (via is WWC23_Players) - [e2 IS WWC23_Pass_Transactions] ->
           (dst is WWC23_Players)
     where e.match_id = 3904629
       and e.match_id = e2.match_id
       and (e2.event_index - e.event_index) < 4
    COLUMNS ( via.player_id, via.player, via.team)
  )
GROUP BY player_id, player, team
ORDER BY Num_In_Middle
DESC FETCH FIRST 10 ROWS ONLY;
```

You can find players involved in a chain of passes

where e.match_id = 3904629

Specific Match

and e.match_id = e2.match_id

Ensure the passes are the same match

and (e2.event_index - e.event_index) < 4

The events happen close to each other

Add logo

You can find players involved in a chain of passes

```
SELECT player_id, player, team, COUNT(1) AS Num_In_Middle
   FROM graph_table (….)
   ) GROUP BY player_id, player, team ORDER BY Num_In_Middle DESC FETCH
FIRST 10 ROWS ONLY;
```

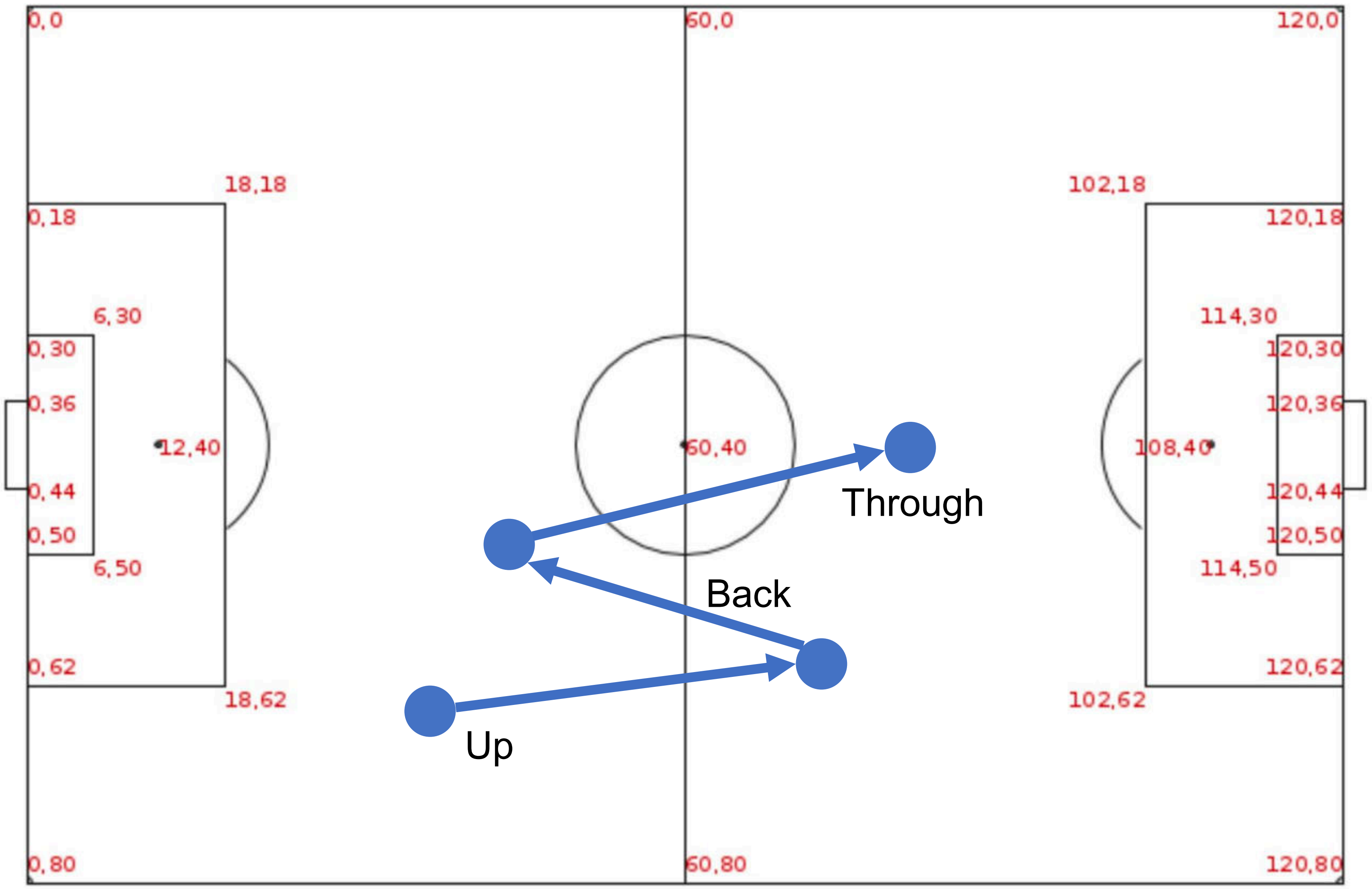
PLAYER_ID	PLAYER	TEAM	NUM_IN_MIDDLE
10252	Alex Greenwood	England Women's	2578
4642	Millie Bright	England Women's	2401
4658	Keira Walsh	England Women's	1712
5000	Stephanie-Elise Catley	Australia Women's	1508
19422	Jessica Carter	England Women's	1182
4643	Georgia Stanway	England Women's	997
5058	Rachel Daly	England Women's	944
5095	Ellie Madison Carpenter	Australia Women's	819
47521	Alessia Russo	England Women's	687
10178	Lucy Bronze	England Women's	676

Add logo

Passing pattern: Up, Back, through

Add logo

Defining an "up, back, through"



Add logo

Find it in our graph

```
SELECT distinct ev_id, ev_id2, ev_id3, period, minute, second, team,
            srcID, srcplayer, x1,   viaID, viaPlayer, x2 dstID, dstPlayer, x3 dstID2, dstPlayer2, x4, (x4-x1) as total_distance
 FROM graph_table ( WWC23_Passing_Graph
MATCH (src is WWC23_Players) - [e IS WWC23_Pass_Transactions] -> (via is WWC23_Players) - [e2 IS WWC23_Pass_Transactions] -> (dst is
WWC23_Players) - [e3 IS WWC23_Pass_Transactions] -> (dst2 is WWC23_Players)
    where e.match_id = 3904629  and e.loc_x < e.end_loc_x
    and e.pass_recipient_id = e2.player_id and e2.pass_recipient_id = e3.player_id
    and e.period = e2.period and e.period = e3.period
    and e.match_id = e2.match_id and e.match_id = e3.match_id
    and e.event_index < e2.event_index and e2.event_index < e3.event_index
    and (e3.event_index - e.event_index) < 10
    COLUMNS (e.event_index as ev_id, e.period, e.minute, e.second,
        src.player_id as srcID, src.player as srcplayer, src.team as team,
        e.loc_x as x1, e.end_loc_x as x2,
        via.player_id as viaID, via.player as viaPlayer,
        dst.player_id as dstID, dst.player as dstPlayer,
        e2.event_index as ev_id2, e2.end_loc_x as x3,
        dst2.player_id as dstID2, dst2.player as dstPlayer2,
        e3.event_index as ev_id3, e3.end_loc_x as x4
        )
    ) order by minute, second asc;
```

Add logo

Find it in our graph

```
and e.loc_x < e.end_loc_x
and e2.end_loc_x < e.end_loc_x
```

"Up, Back…"

```
and e3.end_loc_x > e2.end_loc_x
and src.player_id != dst.player_id
```

"…through"

```
and (e2.event_index -
e.event_index) < 10
```

…that are linked passes

What we found: up, back, through

	EV_ID	EV_ID2	EV_ID3	PERIOD	MINUTE	SECOND	TEAM	SRCID	SRCPLAYER	X1	VIAID	VIAPLAYER	DSTID	DSTPLAYER	DSTID2	DSTPLAYER2	X4
1	24	27	30	1	0	19	England Women's	31538	Mary Alexandra Earps	15.9	10252	Alex Greenwood	21.2	Millie Bright	24.5	Lauren Hemp	87.
2	45	47	49	1	1	9	Australia Women's	401635	Clare Hunt	58.4	4979	Katrina Gorry	67.9	Clare Hunt	53.6	Clare Elizabeth Polkinghorne	45.
3	84	86	88	1	2	41	Australia Women's	401635	Clare Hunt	20.4	5095	Ellie Madison Carpenter	27.8	Katrina Gorry	29.3	Hayley Emma Raso	41.
4	86	88	90	1	2	41	Australia Women's	5095	Ellie Madison Carpenter	28.4	4979	Katrina Gorry	29.3	Hayley Emma Raso	41.5	Ellie Madison Carpenter	38.
5	88	90	96	1	2	43	Australia Women's	4979	Katrina Gorry	29.5	6818	Hayley Emma Raso	41.5	Ellie Madison Carpenter	38.9	Mary Boio Fowler	58.
6	96	98	100	1	3	7	Australia Women's	5095	Ellie Madison Carpenter	50.3	35693	Mary Boio Fowler	58.9	Ellie Madison Carpenter	51.2	Hayley Emma Raso	68.
7	113	116	119	1	3	25	England Women's	4642	Millie Bright	26.5	4643	Georgia Stanway	44.8	Jessica Carter	27.9	Millie Bright	34.
8	122	125	128	1	3	37	England Women's	4642	Millie Bright	38.9	19422	Jessica Carter	44.4	Millie Bright	38.3	Alex Greenwood	39.
9	128	131	134	1	3	44	England Women's	4642	Millie Bright	39	10252	Alex Greenwood	39.5	Ella Toone	62.6	Rachel Daly	57.
10	131	134	136	1	3	47	England Women's	10252	Alex Greenwood	43.1	31534	Ella Toone	62.6	Rachel Daly	57.5	Lauren Hemp	75.
11	141	145	149	1	4	5	Australia Women's	5095	Ellie Madison Carpenter	45.7	4961	Samantha May Kerr	66.2	Katrina Gorry	60.7	Hayley Emma Raso	69.
12	156	159	161	1	4	14	Australia Women's	401635	Clare Hunt	33.7	131586	Kyra Lillee Cooney-Cross	42	Stephanie-Elise Catley	38.6	Caitlin Jade Foord	72.
13	167	169	172	1	4	31	England Women's	4643	Georgia Stanway	39	10178	Lucy Bronze	51.2	Jessica Carter	36.4	Millie Bright	33.
14	175	178	180	1	4	38	England Women's	4642	Millie Bright	34.4	19422	Jessica Carter	39.8	Alessia Russo	87.3	Lauren Hemp	10
15	184	188	191	1	4	53	Australia Women's	42787	Mackenzie Arnold	12	401635	Clare Hunt	20.3	Mackenzie Arnold	7.8	Hayley Emma Raso	64.
16	193	197	200	1	5	2	England Women's	10252	Alex Greenwood	55.3	31534	Ella Toone	69.7	Rachel Daly	63.9	Ella Toone	55.
17	211	214	217	1	5	16	England Women's	19422	Jessica Carter	22.6	4642	Millie Bright	26.7	Jessica Carter	35.6	Millie Bright	34.
18	214	217	220	1	5	20	England Women's	4642	Millie Bright	29.8	19422	Jessica Carter	35.6	Millie Bright	34.8	Alex Greenwood	3
19	220	224	227	1	5	27	England Women's	4642	Millie Bright	36.2	10252	Alex Greenwood	39	Millie Bright	32.9	Rachel Daly	51.
20	247	250	254	1	5	48	England Women's	31538	Mary Alexandra Earps	16.7	4642	Millie Bright	26.4	Jessica Carter	31.1	Mary Alexandra Earps	9.

What we found: up, back, through

EV_ID	EV_ID2	EV_ID3	PERIOD	MINUTE	SECOND	TEAM	SRCPLAYER	X1	VIAPLAYER	X2	DSTPLAYER	X3	DSTPLAYER2	X4	TOTAL_DISTANCE
131	134	136	1	3	47	England Women's	Alex Greenwood	43.1	Ella Toone	62.6	Rachel Daly	57.5	Lauren Hemp	75.8	32.7

Add logo

What we found: up, back, through

```
select team, count(*) as num_UPBACKTHROUGH
From (SELECT distinct ev_id, ev_id2, ev_id3, period, minute, second, team,
            srcID, srcplayer, x1,   viaID, viaPlayer, x2 dstID…..



) group by team;
```

	TEAM	NUM_UPBACKTHROUGH
1	Australia Women's	30
2	England Women's	49

Add logo

Key takeaways

- There is hidden value in unstructured data, but you mind need to think outside the box to get it
- Data-informed is good - but ensure you add your context
- It’s one element of the picture – don’t just rely on it!
- It’s being used across the sports industry, but equally can be used in business, supply chain and finance

Add logo

Ideas and Questions always welcome....

Dr. Abi Giles-Haigh
Oracle Analytics and Innovation Director, Capgemini

Abi.Giles-Haigh@Capgemini.com
AbiHaigh@Hotmail.com
@Abi_Giles_Haigh