THE CHANCE OF A CHANCE

Predicting Goal Attempts Using Football Match Event Sequences

Edward Hine
MSc Data Science Project 2021

Academic Declaration

I have read and understood the sections of plagiarism in the College Policy on assessment offences and confirm that the work is my own, with the work of others clearly acknowledged. I give my permission to submit my report to the plagiarism testing database that the College is using and test it using plagiarism detection software, search engines or meta-searching software.

1. Abstract

This project applies modern data science and machine learning techniques to analyse aspects of the game of association football. The enduring appeal of the beautiful game can perhaps be ascribed to the simplicity, synchronicity, and speed of movements by players creating goal scoring opportunities. Therefore, the specific objective of the project, or 'goal' if you will, is to construct and compare data-driven models that determine a team's likelihood of creating a goal attempt given a sequence of match events within a given possession in a game. This output probability will be referred to as '**the chance of a Chance**'.

The report sets the scene for the subject matter and then goes on to outline the project objectives before describing the toolkit utilised in the analysis. Following this, the nature of the data and the processing it requires is detailed along with some assumptions made. The diverse methodologies applied to the data using the toolkit are reviewed before the specifics of the implementation are presented. The results of the research are then shared, and conclusions drawn. Finally, I refer to areas for future investigation that have occurred to me during the course this project.

The high-level goals of the project were as follows.

- Identify and derive relevant, meaningful features from match event datasets
- Implement a framework to train multiple models to classify instances of the match features for comparative purposes
- Implement visualisations of model performance and results

Table of Contents

2. Introduction

"Football is the ballet of the masses."

Dmitri Shostakovich

What is the point of football? Looking beyond the simplistic explanation of football as an entertaining ball game played between two teams of eleven players on a pitch, I reframed the question to consider: “What does football mean to me?”

To the untrained eye, football can appear chaotic. To the initiated however, there is beauty in this apparent chaos; football is a game of moments, a game of flow, a game of stops and starts, of tempo and crescendo. The goal is not always the goal. The game appears entirely predictable until it is not. The synchronicity of eleven, distinct, human individuals, their alignment of thought, their adherence to agreed strategy or successful discord, culminating in those rare goals; ***that*** is the beautiful game.

The sequences of movements and interactions between players, even if they do not result in a goal, are what the crowd appreciates. Shortly after the groan associated with the missed chance you will hear the reassurance of applause for the mere creation of the opportunity.

Can data science deconstruct the raw data of these synchronous movements and identify patterns in the chaos? Certain assertions are tested during this project which attempt to demonstrate that the application of data science techniques from various domains can indeed reveal interesting insights. More specifically, the project aims to verify whether sequences of recorded movements and interactions can be used to model the probabilities of creating goal-scoring chances.

2.1 Football Through the Ages

Football is a team sport whereby two teams of eleven players compete to move a spherical ball around a pitch and into the opposition goal. The modern version of the game began in the mid 19th Century in England however, the game has a rich background.

Cuju[1]

[1] https://admin.tradingdraft.com/football/cuju-earliest-form-football/

The earliest examples of team, ball-based games date from over 3,000 years ago in Mesoamerican culture[2]. A version called Tchatali was played by the Aztecs and notably played with a rubber ball. "In some ritual occasions, the ball would symbolize the sun and the captain of the losing team would be sacrificed to the gods." Nowadays, the captain of a defeated Premiership side is more likely to be sacrificed to the tabloid press.

Kicking a ball has long been considered entertaining in human history. Cuju in China from the 3rd Century, Marn Gook played by indigenous Australians, Kemari played in Japan and Harpastum played by Roman soldiers can all be considered early versions. The Romans are credited with the introduction of football to Britain. The evolution of the game across Europe saw it periodically banned for violence and the modern game continues to excite passions. Football, as played today, has evolved from attempts at formalisation of the game in the mid-1800s. Key features of the game; no hands and standardisation of ball size and weight, were agreed in London in 1863 when the first Football association was formed. As the game became more established in Britain, it spread further resulting in the global phenomenon we observe today.

2.2 Football as an Industry

Association football is the most popular sport[3] in the world today, with FIFA World Cup matches attracting global audiences of billions[4] and more than 250 million regular active participants[5]. Recent years have seen the rapid development and application of advanced data analytics to deepen understanding of the game, particularly at the highest levels. The capture, analysis and presentation of detailed data pertaining to the professional game has been industrialised. According to Kearney, "Football remains king: Global revenues for this sport equal €20 billion ($28 billion) yearly—almost as much as the combined €23 billion ($32 billion) in revenues for all U.S. sports, Formula 1 racing, tennis and golf.".[6]

2.2.1 "What are the odds?" Gambling and the Game

Gambling has been a key driver to much of the analytics evolution of the beautiful game. While it is an area fraught with ethical dilemma, it is a fact that cutting edge analytics of the game have come in large part from betting industry investment. Recent changes to sponsorship aim to minimise the proliferation of addiction to gambling. However, the scale of the commerce involved means that, while gambling may be hidden, it is not going anywhere. The reference to gambling in the context of this project is to acknowledge that the work done here may offer some potential practical application in this domain.

[2] https://www.footballhistory.org

[3] https://mastersoccermind.com/17-reasons-why-soccer-is-the-most-popular-sport-in-the-world/

[4] https://www.fifa.com/worldcup/news/more-than-half-the-world-watched-record-breaking-2018-world-cup

[5] https://www.fifa.com/who-we-are/news/fifa-survey-approximately-250-million-footballers-worldwide-8804

[6] https://www.de.kearney.com/communications-media-technology/article?/a/the-sports-market

3. Objectives

"In the beginning there was chaos, and football was without form."

Jonathan Wilson

The summary aim of this project will be to evaluate the probability of creating an opportunity for a goal attempt. That is, the intention is to quantify, for a given sequence of match event data, the chance of creating a Chance. The objectives of the project are to explore the assertions listed below. Aspects of each will be explored and then the evaluation presented in the Results and Conclusion sections.

3.1 Speed & Precision

Chance creation is dictated by Speed and Precision in patterns of play

The team interactions and patterns of play that are so entertaining to watch can be deconstructed into their fundamental data points namely pass and carry speeds and distances. One of the basic aims of the project is to see whether these datapoints alone can be modelled to predict Chance creation.

3.2 Using RNNs

Recurrent neural networks can learn to classify the patterns of play that lead to Chance creation

Sequences of numerical data describing match events can be (i) suitably segmented, (ii) labelled and then (iii) modelled for classification tasks using known Recurrent Neural Network architectures and techniques.

3.3 Event sequences as text

Patterns of play can be transformed into descriptive text and NLP models can learn to classify the texts that describe Chance creation

Sequences of numerical data describing match events can be encoded into English text that can be (i) viewed as human-legible text commentary and then (ii) analysed using known NLP techniques, like those that are used to perform document sentiment analysis.

3.4 Event sequence xG regressions

Expected Goal (xG) values can be predicted by a regression of the data describing the patterns of play

As xG is a measure of Chance quality, sequences of data describing match events can be (i) assigned a reasonable xG value depending on their location, and (ii) modelled by a regression to predict an xG value for patterns of play. Thus, the problem can be transformed from one of Chance classification into a regression of Chance quality on sequences of data describing match events.

4. Machine Learning Toolkit

This section outlines a selection of machine learning techniques that are applicable to the tasks at hand, namely classification and regression of sequences of match event data. Suitable models are described and some key aspects of the approach to using them are outlined.

1.1 Training, Validating and Testing

"Everything is practice."

Pele

Models compute parameterised functions that attempt to describe relationships between attributes in a dataset. The process of systematically determining the values of these function parameters is known as training. This training process requires access to a sufficiently sized sample dataset of example observations. In the case of supervised learning, each example observation is appropriately labelled or valued to measure the training error. Once the training process is complete, as determined by convergence or exhausted resources, the model should be able to form reasonable predictions about new unseen data by application of the previously learned parametrised function.

A small subset of the sample dataset, known as the validation dataset, is excluded from the training process. This validation dataset is used to assess model performance periodically after iterations of the function parameters during the training process. This is useful for monitoring any potential bias or variance present in the model during training.

A second subset, the test dataset, is also excluded from the training process. This test dataset is used to assess the final model performance once the function parameters are determined upon completion of the training process. Both the validation and test datasets are "unseen" by the model during the training process, thus providing unbiased estimations of the learned predictive ability of the model.

To facilitate this approach, the sample match event dataset will be divided into three sub-sets as described in the following proportions:

- 72% for model training
- 8% for model validation after each training epoch
- 20% for final model performance testing

4.1 Logistic Regression

Logistic regression models the probability that a response variable Y belongs to one of two classes based upon the input of one or more associated predictor variables, X. The model employs the logistic (sigmoid) function to force the output of a linear model into a probability P between zero and one.

$$P = \frac{e^{(\beta+\alpha_1 X_1+\ldots\alpha_n X_n)}}{1+e^{(\beta+\alpha_1 X_1+\ldots\alpha_n X_n)}}$$

A maximum likelihood method is used to fit the model to the training data. The coefficients are chosen to maximise the correspondence between the probability of each observation belonging to a class and the actual observed class. In the context of the project, this probability would correspond to the chance of a Chance. For given selected event data X, the ideal logistic function will output a '1' when the event data produces a Chance, and a zero otherwise.

However, given that it employs the above sigmoid function, Logistic regression is not amenable to modelling variable length sequences of data, such as the match events here. Hence the data needs to be manipulated into a vector format through a process of data consolidation before any fitting. This may reduce the predictive power of the model but will provide a good baseline against which to measure more complex models.

4.2 Neural Networks

Neural Networks have become near ubiquitous in modern solutions to computing and data science problems in every domain. They provide a flexible method for learning underlying patterns and relationships in datasets. There are many mature, third-party software packages that make the implementation and deployment of Neural Networks more streamlined. Example packages are TensorFlow and Keras both of which feature in my implementation.

The basic structure of a neural network comprises several interconnected layers of units called artificial neurons, analogous to neurons found in biological brains. Numeric signals are propagated between connected neurons which start with random 'connection weights' that qualify the signals. The weighted inputs are mapped through a specified activation function to determine each unit's output state. The connection weights are then repeatedly adjusted according to a specified output error metric propagated backwards during the training phase of the network.[7]

The many interconnections between one or more so-called "hidden" layers of neurons provide the network with an ability to learn any complex, non-linear patterns within datasets of sufficient size.[8] Given the volume and complexity of the match event data, neural networks are an appropriate choice of model. It is critical that an appropriate network architecture is chosen for the task at hand.

4.3 Recurrent and LSTM Networks

The high-level problem to be addressed in this project is that of sequence classification[9]; that is,

[7] http://neuralnetworksanddeeplearning.com/chap2.html

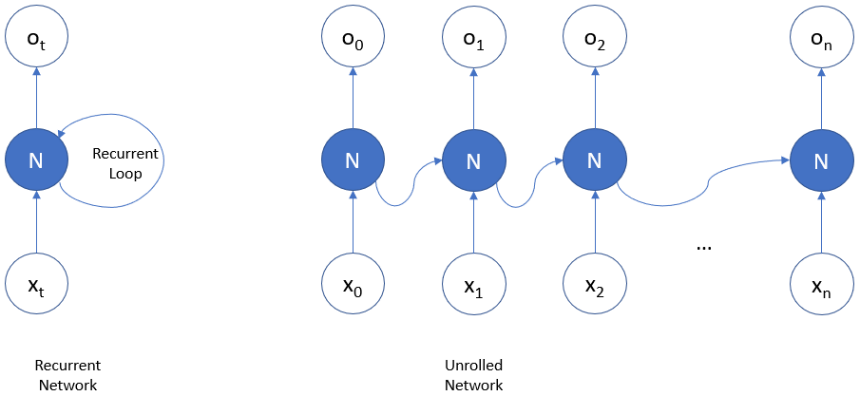
[8] http://neuralnetworksanddeeplearning.com/chap4.html

[9] https://machinelearningmastery.com/sequence-classification-lstm-recurrent-neural-networks-python-keras/

constructing a model that can learn patterns present in sequences of inputs and then successfully predict a class for previously unseen test sequences. In the Statsbomb dataset used, the sequences of inputs are of variable length and comprise numeric and text-labelled features, such as length or speed, that describe the series of events that occur in each possession in a match.

Therefore, within each input sequence, the model should have the capacity to learn any long-term patterns and dependencies between events. Recurrent neural networks (RNNs), and Long Short-Term Memory (LSTM) networks are especially suitable for this purpose (1997 Hochreiter & Schmidhuber[10]).

RNNs are built in a similar fashion to basic Neural Networks but with additional chain-like structures that provide feedback loops within the network. This structure enables information from previous steps in input sequences to persist (recur) within the network as later parts of the sequence are subsequently input. In fact, it can be seen that "unrolling" the recurrent loop within the RNN reproduces the structure of a basic neural network, with signals now able to pass from one iteration of input to the next. Below, input x at time t is transmitted to network segment N to produce output o. The unrolled version depicts how the signal from time t=0 is transmitted into the next iteration at time t=1 and so on.



Unrolling the recurrent loop of an RNN

RNNs have proved particularly useful in fields such as Natural Language Processing and Machine Speech Recognition among others[11]. However, in their base form as described above, they can be limited in their ability to capture longer term dependencies within sequences. An RNN employed to analyse text as described would be expected to learn dependencies between adjacent (or at least nearby) words in a sentence but would be unable to capture any dependencies between words that are further apart in longer sentences.

For example, here are two sentences that demonstrate how the dependencies between words in sentences may vary with the length, with the target word (to be predicted) in bold:

[10] www.bioinf.jku.at/publications/older/2604.pdf

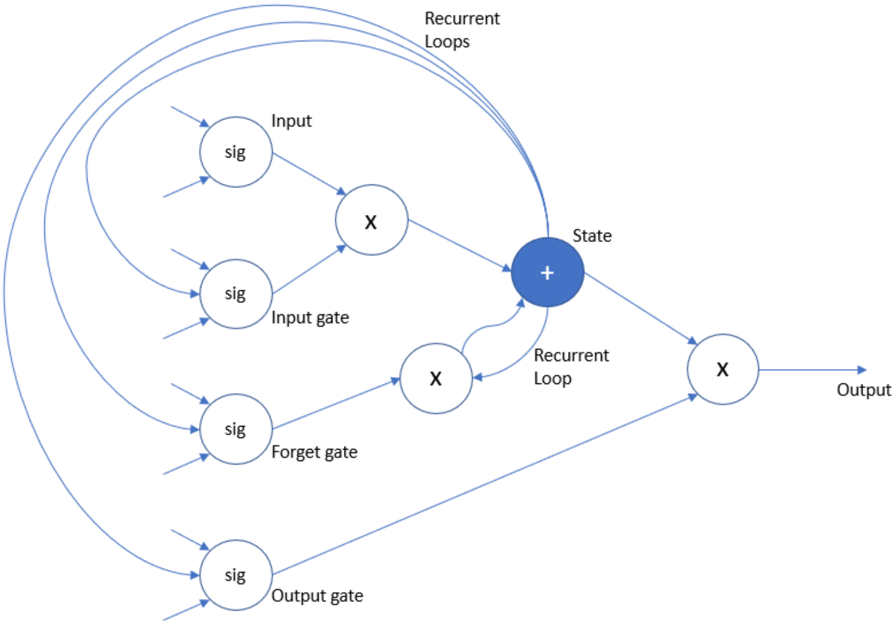
[11] https://karpathy.github.io/2015/05/21/rnn-effectiveness

(1) *I'm in the away supporter's end watching the* ***match****.*

(2) *As a football fan native to North-East Lincolnshire, while I have lived in London for over twenty years, my heart belongs to* ***Grimsby Town****.*

A basic, pre-trained RNN would likely be able to detect the short-term dependencies present in sentence (1) and therefore be able to predict the word "match" given the preceding sequence of words. However, due to the larger gap between the dependent words in sentence (2), the simple RNN would be unable to capture the long-term dependency and therefore struggle to predict "Grimsby Town" based on the subtle clues at the start of the sentence.

Networks built using LSTM units are designed to capture these long-term contexts and dependencies by employing specialised cells that retain memory of earlier elements within long sequences. Each LSTM unit has a persisting long-term state that is manipulated by four embedded sub-units that implement the memory infrastructure.



Internals of an LSTM Unit

Three of the sub-units are gate controllers that are operated by a logistic activation function (i.e., output a value between 0 and 1) denoted as **sig**(moid) in the diagram, with the fourth accepting the input from the previous step's short-term state. The multiplicative combination of the three fully connected gated controllers and the short-term input layer determines at each step what information is retained in long-term memory, what is forgotten, and what is output as short-term state to be passed to the next step.

This architecture enables a LSTM cell to add and retain any important input to the long-term state, eject any irrelevant input as required, and still retrieve it as needed. This ability enables LSTM-based networks to capture the long-term dependencies in sequences as described.

This analysis suggests that LSTM units would be an appropriate choice to use in constructing a Neural

Network to model the sequences of events in football matches.

4.4 Convolutional Networks

A Convolutional Neural Network[12] employs one or more Convolutional layers within its structure. The purpose of the convolutional layer is to filter, summarise and preserve a valuable aspect of information from the previous layer.

A CNN is traditionally used in the domain of image processing where it can be used to help identify macro-level features present in the image without reference to the specific location within the overall image. For example, an initial convolution would identify lines or edges within an image, and a subsequent convolution would identify basic shapes constructed from those lines. This helps to form a type of hierarchy of increasingly higher-level features which is appropriate for working with images and image recognition tasks.

Pooling[13] layers are typically used in addition to convolutional layers to further reduce the sample size of an input layer, reducing computational costs, the number of parameters, and thus the likelihood of overfitting. Frequently the pooling techniques used are simply to map a specified number of input features to either an average value or a maximum value of the inputs in the pool size – so-called average pooling or max pooling.

To analyse football match event sequences, a convolutional layer and a pooling layer will be used to identify the latent structures and dependencies in the text-based sequences.

4.5 Network Training Optimisers - WAME

Neural networks can be trained more efficiently through the application of optimisation techniques, specifically training algorithms. The paper 'Training Convolutional Networks with Weight–wise Adaptive Learning Rates'[14] by Mosca and Magoulas describes one such training algorithm, Weight-wise Adaptive Moment Estimation, abbreviated to WAME. WAME has been shown to improve convergence speed and offers similar or better benchmark accuracy compared to other well-established optimisation techniques such as Adam, RMSProp or SGD.

Having implemented a version of WAME using Tensorflow2.0 during my Birkbeck Machine Learning coursework assignment, I have employed it again here to improve the training performance of the LSTM network.

4.6 Assessing Model Performance

This project employs two general approaches to model Chance creation: Classification and Regression. Both approaches require the specification of suitable methods, called metrics, to quantify their

[12] https://machinelearningmastery.com/cnn-models-for-human-activity-recognition-time-series-classification/

[13] https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/

[14] Training Convolutional Networks with Weight–wise Adaptive Learning Rates Alan Mosca and George D. Magoulas (M&M)

respective performance.

4.6.1 Classification Metrics

Classification models seek to correctly assign one of two labels, in this case; Chance or No chance, to each example match event sequence. Several metrics may be assessed to measure efficacy in this approach as below:

- **Binary Accuracy** – a percentage value that summarises the ratio of model predictions that correctly classify the match event sequences with the correct underlying label.
- **False Positives** – examples of match event sequences that are incorrectly predicted to result in Chances by the model.
- **False Negatives** – examples of match event sequences that are incorrectly predicted to not result in Chances by the model.
- **Confusion Matrix** – a summary grid of cumulative counts of model predictions, cross-referencing Chance class ground truth values.

4.6.2 Regression Metrics

Regression models seek to assign a real value, in this case, xG, to each example match event sequence. Several metrics may be applied to measure success in this approach:

- **Mean Squared Error** – the mean of the square of the differences between the n xG values predicted by the model, pred(y_n), and the underlying xG values, y_n, associated with each match event sequence.

$$\text{MSE} = \frac{1}{n}\sum_{1}^{n}(pred(y_n) - y_n)^2$$

- **R-Squared** (R^2 or R2) Score – measures the proportion of the variance in the xG value (dependent variable) that the model can predict using the match event sequences (independent variables). An ideal R-Squared value is 1, representing perfect correlation between the actual xG and predicted xG calculated using the input match event sequence data. A lower value approaching zero would indicate that the model is not suitable for purpose and has little predictive ability.

4.7 The Bias/Variance Trade-off

Models naturally introduce some form of error when they are approximating a real-world situation. Models must therefore strike a balance between simplicity and complexity to ensure the error levels in their output are acceptable.

Error introduced to the model ***by bias*** arises when a model is too simplistic and inflexible with respect to the training data. That is, the model will not be able to capture enough of the underlying patterns in the data. This situation is what is known as “underfitting” the training data. Predictive performance on training and unseen test data will therefore suffer. Bias error can be reduced by introducing additional variables, and thus flexibility, to the model.

Error introduced ***by variance*** arises when the model is too sensitive and flexible with respect to the training data. The model constrains its parameters to the specifics of the training data at the expense of generalising ability. This situation is what is known as “overfitting” the training data. Predictive performance using unseen test data will therefore suffer. Variance error can be reduced by using additional training data or simplifying the model.

To combat both bias and variance error arising in the models used in this project, the training phase was monitored closely. Training performance can be recorded and visualised. If training accuracy is not stabilising at sufficiently high levels, then this may be an indication that the model has high bias and is underfitting the training data. If the training accuracy and validation accuracy begin to diverge, this may be an indication that the model has high variance and is overfitting the training data. In either case, the number of training epochs can be adjusted accordingly.

Additionally, to help combat overfitting, a neural network can be configured with a so-called “dropout” layer to regularise the training performance. This technique is explored in more detail in the Implementation section.

5. Data

"We must have had 99 percent of the game. It was the other three percent that cost us the match."

Ruud Gullit

5.1 Statsbomb Open Data

The granularity of the data required for this analysis of Chance creation is quite specific. StatsBomb[15] [16] is a football analytics provider covering multiple leagues around the globe, including analytics provisions for more than half of the English Premier League teams. The data they provide is comprehensive and at a level of granularity suitable for purpose. A subset of their total data offering is freely available for public use, on the condition of a credit and inclusion of their logo (below).

STATSBOMB

INSIDE EVERY PASS, EVERY SHOT, EVERY MOVE

The free dataset they provide is extensive, covering matches, team line-ups and match events from several competitions including several years of Spanish LaLiga, Champions League, Men's FIFA World Cup 2018 and the Women's FIFA World Cup 2019. The data captured is detailed and provided conveniently in JSON format or via a basic Python API that provides unified access to each flavour of data.

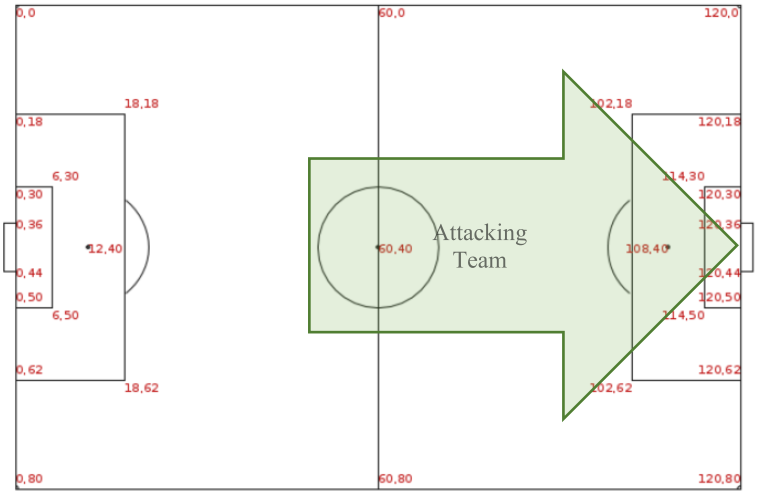
The extraction process is described later in this section. Once the extraction is completed, the event records for each available match are consolidated in a single Pandas DataFrame. Each individual event record is described by a set of 86 attributes not all of which are utilised in my analyses. Some of these are shown in the tables that follow. However, many attributes of interest contain some reference to the pitch location. I will elaborate later upon the pivotal role played by pitch location in the dataset. Some post-extraction processing is required to prepare the dataset for further analysis, and this is detailed in the 'Data Preparation Process' section.

[15] https://statsbomb.com

[16] https://github.com/statsbomb/open-data

5.2 Pitch Location Map

The pitch location map, illustrated below, depicts the coordinate system used by the StatsBomb data provider to label their match event data with reference to the location on the pitch.



Statsbomb Reference Pitch Co-ordinates

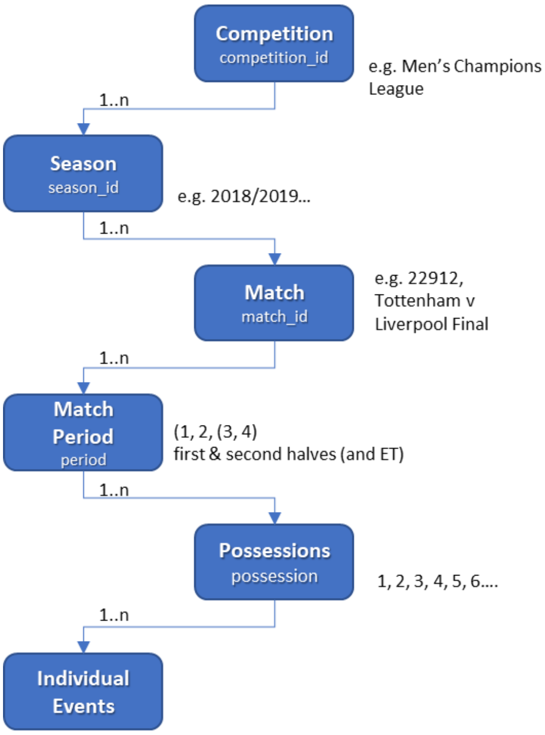
On the diagram above, the team in possession of the ball is always trying to score in the right-hand side goal at location [120, 40]. The coordinate system is effectively mirrored and then overlaid; so both teams are targeting the goal at x=120. If a Team Blue striker shoots from location (100,40) then the Team Red goalkeeper that makes the save may well be at location (5,40) and therefore ***not*** at (115,40)).

Put another way, most defensive events, regardless of which team performs them, will occur in locations of x<~30 while most attacking events will occur in locations of x>~90.

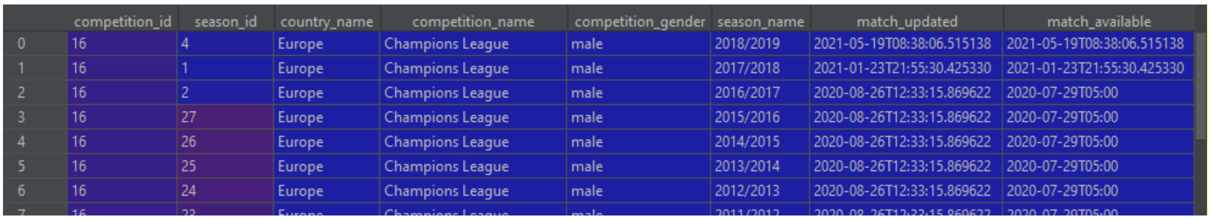
In real football stadiums, there is some variance in pitch size from ground to ground. This variation is allowed within the regulations of the game. Variance in pitch size is accommodated by StatsBomb and data is standardised in this regard.

5.3 Data Preparation Process

Statsbomb data is organised in a hierarchical structure.

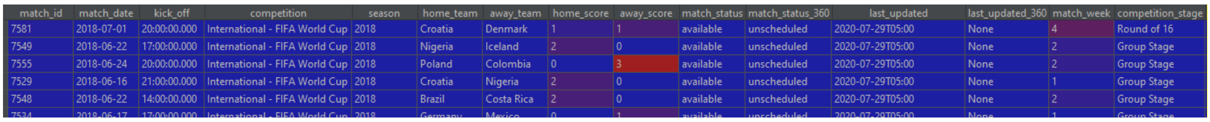


Entity Relationship Diagram of Statsbomb Event Data



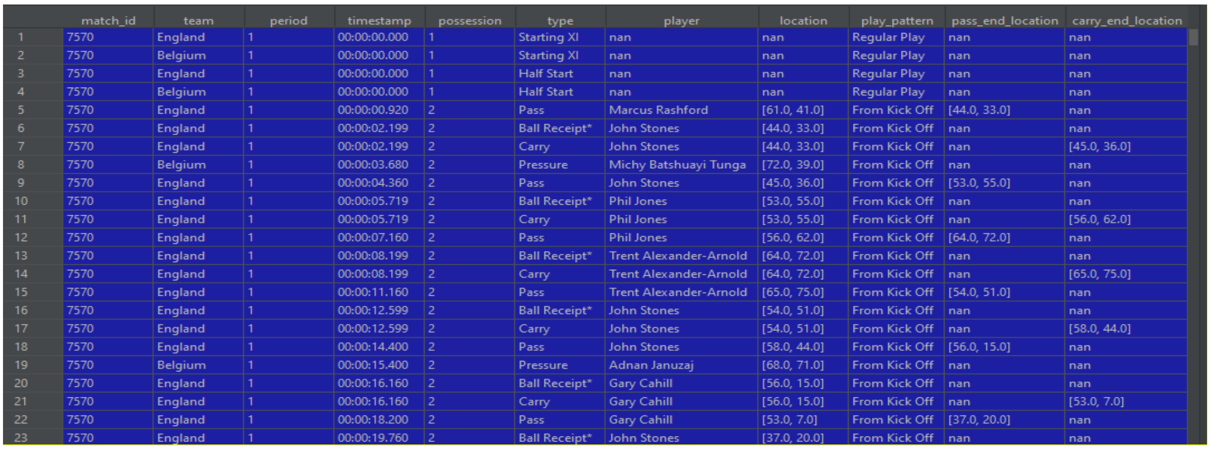
	competition_id	season_id	country_name	competition_name	competition_gender	season_name	match_updated	match_available
0	16	4	Europe	Champions League	male	2018/2019	2021-05-19T08:38:06.515138	2021-05-19T08:38:06.515138
1	16	1	Europe	Champions League	male	2017/2018	2021-01-23T21:55:30.425330	2021-01-23T21:55:30.425330
2	16	2	Europe	Champions League	male	2016/2017	2020-08-26T12:33:15.869622	2020-07-29T05:00
3	16	27	Europe	Champions League	male	2015/2016	2020-08-26T12:33:15.869622	2020-07-29T05:00
4	16	26	Europe	Champions League	male	2014/2015	2020-08-26T12:33:15.869622	2020-07-29T05:00
5	16	25	Europe	Champions League	male	2013/2014	2020-08-26T12:33:15.869622	2020-07-29T05:00
6	16	24	Europe	Champions League	male	2012/2013	2020-08-26T12:33:15.869622	2020-07-29T05:00
7	16	23	Europe	Champions League	male	2011/2012	2020-08-26T12:33:15.869622	2020-07-29T05:00

Examples of available competitions



match_id	match_date	kick_off	competition	season	home_team	away_team	home_score	away_score	match_status	match_status_360	last_updated	last_updated_360	match_week	competition_stage
7581	2018-07-01	20:00:00.000	International - FIFA World Cup	2018	Croatia	Denmark	1	1	available	unscheduled	2020-07-29T05:00	None	4	Round of 16
7549	2018-06-22	17:00:00.000	International - FIFA World Cup	2018	Nigeria	Iceland	2	0	available	unscheduled	2020-07-29T05:00	None	2	Group Stage
7555	2018-06-24	20:00:00.000	International - FIFA World Cup	2018	Poland	Colombia	0	3	available	unscheduled	2020-07-29T05:00	None	2	Group Stage
7529	2018-06-16	21:00:00.000	International - FIFA World Cup	2018	Croatia	Nigeria	2	0	available	unscheduled	2020-07-29T05:00	None	1	Group Stage
7548	2018-06-22	14:00:00.000	International - FIFA World Cup	2018	Brazil	Costa Rica	2	0	available	unscheduled	2020-07-29T05:00	None	2	Group Stage
7534	2018-06-17	17:00:00.000	International - FIFA World Cup	2018	Germany	Mexico	0	1	available	unscheduled	2020-07-29T05:00	None	1	Group Stage

Examples of available matches



	match_id	team	period	timestamp	possession	type	player	location	play_pattern	pass_end_location	carry_end_location
1	7570	England	1	00:00:00.000	1	Starting XI	nan	nan	Regular Play	nan	nan
2	7570	Belgium	1	00:00:00.000	1	Starting XI	nan	nan	Regular Play	nan	nan
3	7570	England	1	00:00:00.000	1	Half Start	nan	nan	Regular Play	nan	nan
4	7570	Belgium	1	00:00:00.000	1	Half Start	nan	nan	Regular Play	nan	nan
5	7570	England	1	00:00:00.920	2	Pass	Marcus Rashford	[61.0, 41.0]	From Kick Off	[44.0, 33.0]	nan
6	7570	England	1	00:00:02.199	2	Ball Receipt*	John Stones	[44.0, 33.0]	From Kick Off	nan	nan
7	7570	England	1	00:00:02.199	2	Carry	John Stones	[44.0, 33.0]	From Kick Off	nan	[45.0, 36.0]
8	7570	Belgium	1	00:00:03.680	2	Pressure	Michy Batshuayi Tunga	[72.0, 39.0]	From Kick Off	nan	nan
9	7570	England	1	00:00:04.360	2	Pass	John Stones	[45.0, 36.0]	From Kick Off	[53.0, 55.0]	nan
10	7570	England	1	00:00:05.719	2	Ball Receipt*	Phil Jones	[53.0, 55.0]	From Kick Off	nan	nan
11	7570	England	1	00:00:05.719	2	Carry	Phil Jones	[53.0, 55.0]	From Kick Off	nan	[56.0, 62.0]
12	7570	England	1	00:00:07.160	2	Pass	Phil Jones	[56.0, 62.0]	From Kick Off	[64.0, 72.0]	nan
13	7570	England	1	00:00:08.199	2	Ball Receipt*	Trent Alexander-Arnold	[64.0, 72.0]	From Kick Off	nan	nan
14	7570	England	1	00:00:08.199	2	Carry	Trent Alexander-Arnold	[64.0, 72.0]	From Kick Off	nan	[65.0, 75.0]
15	7570	England	1	00:00:11.160	2	Pass	Trent Alexander-Arnold	[65.0, 75.0]	From Kick Off	[54.0, 51.0]	nan
16	7570	England	1	00:00:12.599	2	Ball Receipt*	John Stones	[54.0, 51.0]	From Kick Off	nan	nan
17	7570	England	1	00:00:12.599	2	Carry	John Stones	[54.0, 51.0]	From Kick Off	nan	[58.0, 44.0]
18	7570	England	1	00:00:14.400	2	Pass	John Stones	[58.0, 44.0]	From Kick Off	[56.0, 15.0]	nan
19	7570	Belgium	1	00:00:15.400	2	Pressure	Adnan Januzaj	[68.0, 71.0]	From Kick Off	nan	nan
20	7570	England	1	00:00:16.160	2	Ball Receipt*	Gary Cahill	[56.0, 15.0]	From Kick Off	nan	nan
21	7570	England	1	00:00:16.160	2	Carry	Gary Cahill	[56.0, 15.0]	From Kick Off	nan	[53.0, 7.0]
22	7570	England	1	00:00:18.200	2	Pass	Gary Cahill	[53.0, 7.0]	From Kick Off	[37.0, 20.0]	nan
23	7570	England	1	00:00:19.760	2	Ball Receipt*	John Stones	[37.0, 20.0]	From Kick Off	nan	nan

Examples of available events (selected attributes for formatting)

The StatsBomb API is tailored to give access to individual matches; therefore, an initial aggregation of events is required to build a single massive dataset of events across all competitions. This is achieved by repeatedly accessing the API and exhausting the list of available match identifiers and events at each level of the hierarchy.

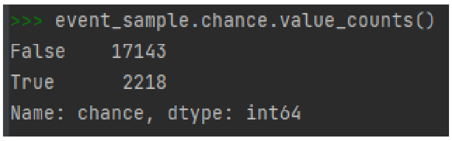
I have built a pre-processing script to aggregate the event data from all available matches into a single Pandas DataFrame, which is then stored locally in a CSV file for ease of access. This avoids the need for the API to access the data remotely over the network. The high-level description of the process is as follows:

1. Extract a list of competitions using the Statsbomb API
2. For each combination of competition and season extract available matches
3. For each match, extract a DataFrame of match events
4. For each period within a match (which usually translates to a match half), process the events as follows:
    a. Split the “location” column into separate values and add “location_x” and “location_y” columns
    b. Add a column to store a Boolean value for Chance, set according to the presence of a valid value in the “shot_type” column
    c. Calculate and add the “carry_length” column, calculated by using Euclidean distance
    d. Calculate and add the “to_goal” column, calculated by using Euclidean distance
    e. Map an xG value from the constructed xG model into the “xg” column
5. Concatenate each batch of events into a Pandas DataFrame and ultimately store as a CSV file

After the script has exhausted the API and processed the extracted data the dataset consists of just over 3 million event records from 878 matches.

5.4 Addressing Class Imbalance

Goal scoring chances are created relatively rarely in a game. Therefore, there exists an imbalance in the dataset between example possession sequences that end in a chance and those that do not.



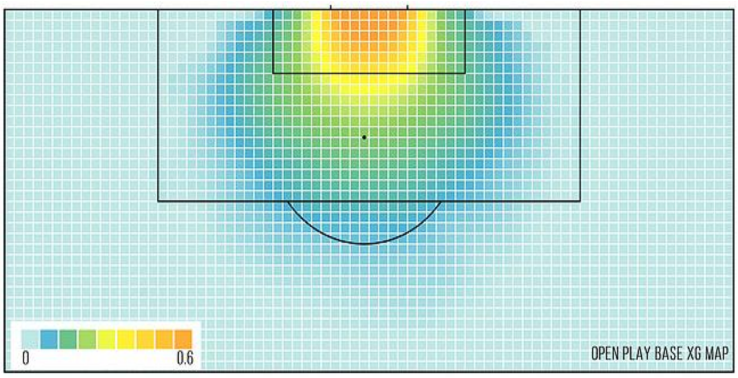
```
>>> event_sample.chance.value_counts()
False    17143
True      2218
Name: chance, dtype: int64
```

Dataframe value_counts operation demonstrating Chance class imbalance of a sample dataset

Model training can be negatively affected by an imbalanced dataset. The presence of fewer examples of the minority class can limit a neural network’s ability to learn the characteristics of that class, especially when training using batches of examples. To mitigate for this, I have chosen to implement an oversampling method. This technique randomly samples, with replacement, duplicate examples of possession sequences in the minority Chance=True class. These are added to the dataset until the numbers of possessions belonging to both Chance classes are evenly balanced.

5.5 [X,Y] Marks the Spot: Location-Data Leakage

Chance creation is highly correlated with pitch location. This obviously follows as the closer the ball is to the goal, the more likely it is that a Chance is created. This is evidenced by examining any xG model heat map, where the increase in scoring probability is clearly highlighted in the area nearest the goal mouth.[17]



As such, the inclusion of any data that relates to pitch location, or allows the model to infer the same, somewhat pollutes the training data. A model will simply learn that the most relevant factor to use in classifying Chances or regressing xG values is the location of the ball relative to the goal. It follows that the model will tend to give less importance to other more interesting patterns in the data - such as the proposed speed of passing or length of carry. This project seeks to demonstrate that these are more interesting critical variables in the creation of Chances, and indeed that it is possible to predict xG values using other more interesting features than location. Hence the leakage of location-specific features in the training dataset must be at least minimised and at best eliminated.

The following examples demonstrate how location data could potentially be allowed to leak into the training data submitted for a model to learn, thus reducing the relevance of any results:

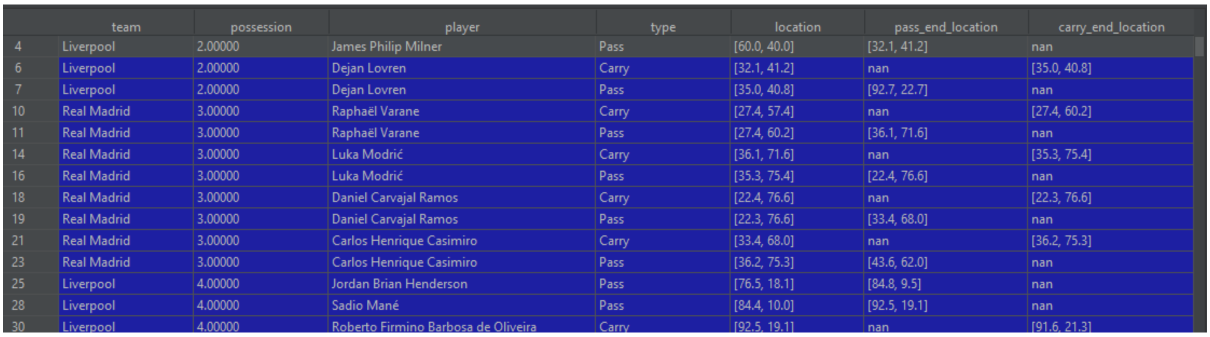
- Including the literal X & Y co-ordinates
- Including text commentary to identify when the ball is in the penalty box
- Including any event data or text that relates to the goalkeeper as the goalkeeper is typically close to the goal

Therefore, training data should, as much as possible, only include features that do not allow specific inference of location. Lengths relating to events are acceptable (pass length, carry length) as they do

[17] https://www.datofutbol.cl/xg-model

not *typically* reveal literal pitch location, only relative locations.

5.6 Derived Movement Data



	team	possession	player	type	location	pass_end_location	carry_end_location
4	Liverpool	2.00000	James Philip Milner	Pass	[60.0, 40.0]	[32.1, 41.2]	nan
6	Liverpool	2.00000	Dejan Lovren	Carry	[32.1, 41.2]	nan	[35.0, 40.8]
7	Liverpool	2.00000	Dejan Lovren	Pass	[35.0, 40.8]	[92.7, 22.7]	nan
10	Real Madrid	3.00000	Raphaël Varane	Carry	[27.4, 57.4]	nan	[27.4, 60.2]
11	Real Madrid	3.00000	Raphaël Varane	Pass	[27.4, 60.2]	[36.1, 71.6]	nan
14	Real Madrid	3.00000	Luka Modrić	Carry	[36.1, 71.6]	nan	[35.3, 75.4]
16	Real Madrid	3.00000	Luka Modrić	Pass	[35.3, 75.4]	[22.4, 76.6]	nan
18	Real Madrid	3.00000	Daniel Carvajal Ramos	Carry	[22.4, 76.6]	nan	[22.3, 76.6]
19	Real Madrid	3.00000	Daniel Carvajal Ramos	Pass	[22.3, 76.6]	[33.4, 68.0]	nan
21	Real Madrid	3.00000	Carlos Henrique Casimiro	Carry	[33.4, 68.0]	nan	[36.2, 75.3]
23	Real Madrid	3.00000	Carlos Henrique Casimiro	Pass	[36.2, 75.3]	[43.6, 62.0]	nan
25	Liverpool	4.00000	Jordan Brian Henderson	Pass	[76.5, 18.1]	[84.8, 9.5]	nan
28	Liverpool	4.00000	Sadio Mané	Pass	[84.4, 10.0]	[92.5, 19.1]	nan
30	Liverpool	4.00000	Roberto Firmino Barbosa de Oliveira	Carry	[92.5, 19.1]	nan	[91.6, 21.3]

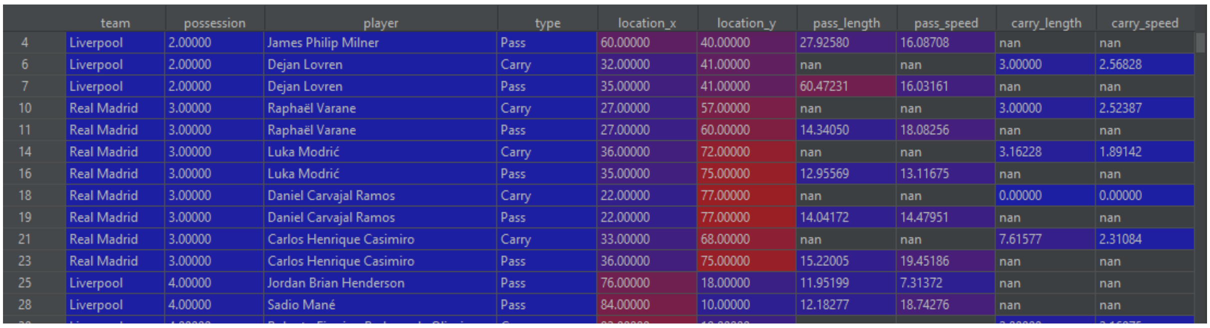
Examples of selected pass and carry event attributes

All relevant events have a location, consisting of X and Y coordinates stored as a list of length two. These are extracted and appended to the match event DataFrame as two additional separate fields to allow the X and Y coordinates to be accessed directly.

Events of interest, specifically passes and carries, have end locations recorded in the form of lists. For efficiency and clarity, the location data for these events is transformed from a list consisting of two values into two separate, additional fields with X and Y coordinates for end locations, named 'pass_end_location' and 'carry_end_location' as appropriate.

The start and end coordinates can then be used to calculate a length associated with each of these events. This represents the straight-line displacement of the ball resulting from the event. This length is calculated using a simple Euclidean distance function and stored as an additional field in the DataFrame named 'pass_length' and 'carry_length' as appropriate.

These event records also have an associated duration field. This can be used in conjunction with the length field to calculate a speed for each pass/carry event. This is stored as an additional field in the DataFrame named 'pass_speed' and 'carry_speed' as appropriate.

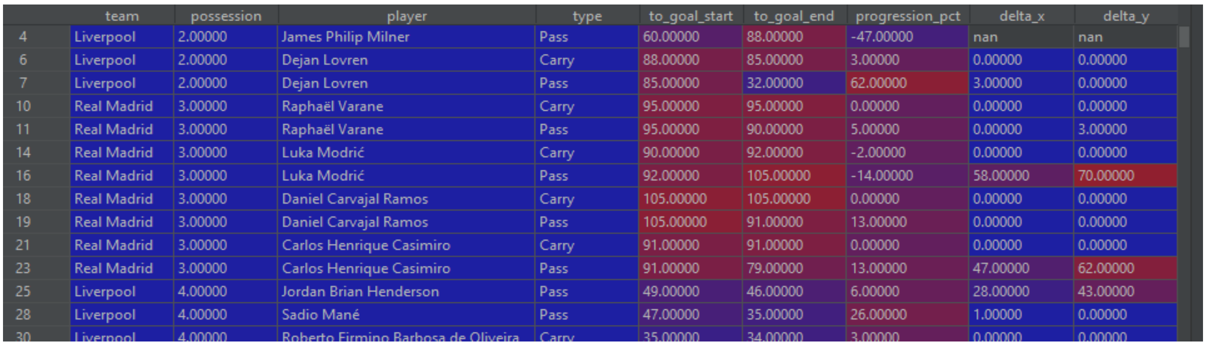


	team	possession	player	type	location_x	location_y	pass_length	pass_speed	carry_length	carry_speed
4	Liverpool	2.00000	James Philip Milner	Pass	60.00000	40.00000	27.92580	16.08708	nan	nan
6	Liverpool	2.00000	Dejan Lovren	Carry	32.00000	41.00000	nan	nan	3.00000	2.56828
7	Liverpool	2.00000	Dejan Lovren	Pass	35.00000	41.00000	60.47231	16.03161	nan	nan
10	Real Madrid	3.00000	Raphaël Varane	Carry	27.00000	57.00000	nan	nan	3.00000	2.52387
11	Real Madrid	3.00000	Raphaël Varane	Pass	27.00000	60.00000	14.34050	18.08256	nan	nan
14	Real Madrid	3.00000	Luka Modrić	Carry	36.00000	72.00000	nan	nan	3.16228	1.89142
16	Real Madrid	3.00000	Luka Modrić	Pass	35.00000	75.00000	12.95569	13.11675	nan	nan
18	Real Madrid	3.00000	Daniel Carvajal Ramos	Carry	22.00000	77.00000	nan	nan	0.00000	0.00000
19	Real Madrid	3.00000	Daniel Carvajal Ramos	Pass	22.00000	77.00000	14.04172	14.47951	nan	nan
21	Real Madrid	3.00000	Carlos Henrique Casimiro	Carry	33.00000	68.00000	nan	nan	7.61577	2.31084
23	Real Madrid	3.00000	Carlos Henrique Casimiro	Pass	36.00000	75.00000	15.22005	19.45186	nan	nan
25	Liverpool	4.00000	Jordan Brian Henderson	Pass	76.00000	18.00000	11.95199	7.31372	nan	nan
28	Liverpool	4.00000	Sadio Mané	Pass	84.00000	10.00000	12.18277	18.74276	nan	nan

Examples of the derived movement data relating to pass and carry events

Each pass/carry event now contains features that record the location at the start of the event, and the location after the event has occurred. The start location can be used to calculate the distance to goal at the start of the event. The end location can be used to calculate the distance to goal after the event has occurred. The ratio of these two distances to goal can be used to calculate a progression percentage. This is calculated and is stored as an additional field in the DataFrame named ‘progression_pct’. Note that progression percentage is potentially a leaky field in terms of the pitch location. Any event with a high progression percentage i.e., one that moves the ball substantially closer to the goal will be highly correlated with Chance creation.

In a similar fashion, the simple delta in X and Y, i.e. the relative displacement of the ball, due to an event is calculated and stored in the DataFrame in ‘delta_x’ and ‘delta_x’ fields. In raw form these values are somewhat leaky in terms of pitch location, especially in respect to the X axis (i.e., the horizontal distance from the goal line). For example, any event that results in the displacement of the ball from a team’s own half to the edge of the opposition box will necessarily increase the likelihood that a chance is created. However, these features may be useful in a) exploring the data, b) sanity checking results and c) defining certain patterns of play upon which I will elaborate further in the ‘Areas for Further Investigation’ section.



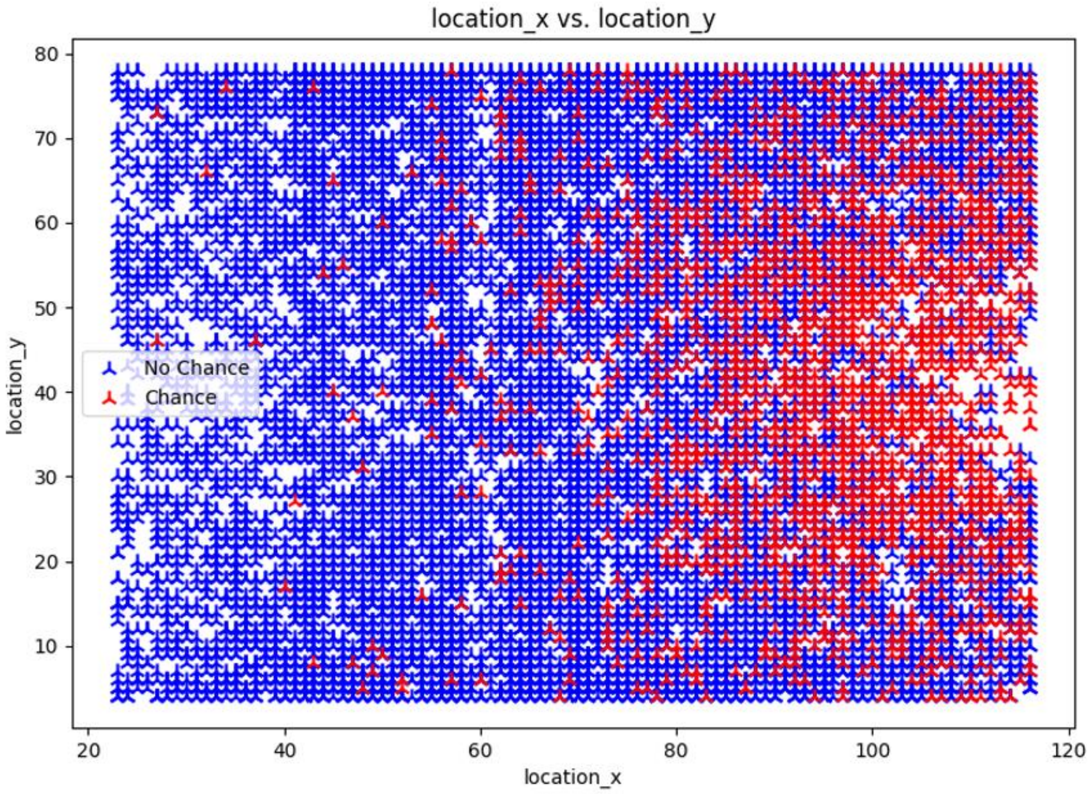
	team	possession	player	type	to_goal_start	to_goal_end	progression_pct	delta_x	delta_y
4	Liverpool	2.00000	James Philip Milner	Pass	60.00000	88.00000	-47.00000	nan	nan
6	Liverpool	2.00000	Dejan Lovren	Carry	88.00000	85.00000	3.00000	0.00000	0.00000
7	Liverpool	2.00000	Dejan Lovren	Pass	85.00000	32.00000	62.00000	3.00000	0.00000
10	Real Madrid	3.00000	Raphaël Varane	Carry	95.00000	95.00000	0.00000	0.00000	0.00000
11	Real Madrid	3.00000	Raphaël Varane	Pass	95.00000	90.00000	5.00000	0.00000	3.00000
14	Real Madrid	3.00000	Luka Modrić	Carry	90.00000	92.00000	-2.00000	0.00000	0.00000
16	Real Madrid	3.00000	Luka Modrić	Pass	92.00000	105.00000	-14.00000	58.00000	70.00000
18	Real Madrid	3.00000	Daniel Carvajal Ramos	Carry	105.00000	105.00000	0.00000	0.00000	0.00000
19	Real Madrid	3.00000	Daniel Carvajal Ramos	Pass	105.00000	91.00000	13.00000	0.00000	0.00000
21	Real Madrid	3.00000	Carlos Henrique Casimiro	Carry	91.00000	91.00000	0.00000	0.00000	0.00000
23	Real Madrid	3.00000	Carlos Henrique Casimiro	Pass	91.00000	79.00000	13.00000	47.00000	62.00000
25	Liverpool	4.00000	Jordan Brian Henderson	Pass	49.00000	46.00000	6.00000	28.00000	43.00000
28	Liverpool	4.00000	Sadio Mané	Pass	47.00000	35.00000	26.00000	1.00000	0.00000
30	Liverpool	4.00000	Roberto Firmino Barbosa de Oliveira	Carry	35.00000	34.00000	3.00000	0.00000	0.00000

Examples of the derived progression and displacement features for pass and carry events

5.7 Some Interesting Plots

The derived movement features, described in the section above, can now be explored using scatter plots. To enable comparison between possession sequences, the derived movement data is aggregated at the match-possession level in appropriate ways as detailed below.

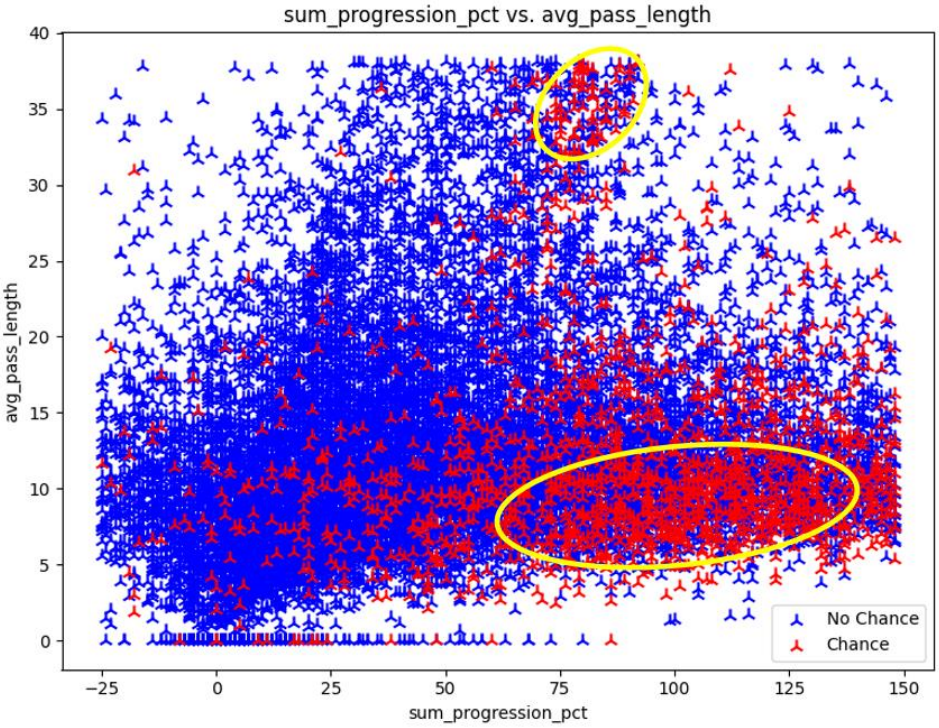
5.7.1 Possession End Location and Chances



This plot depicts a sample of 20,000 possessions. Each point represents the X-Y coordinates of the final pass or carry event in each possession. Specifically, shot events are not included. Each point is then labelled to identify possessions that contain a Chance or No chance. Another way to think about these points is that they typically represent the location of the final pass or carry of a possession when a Chance was created or was not created. This again demonstrates (and sanity checks) the premise that location on the pitch is a key driver in attempts on goal, as clearly the Chances are clustered towards the goal line at x=120.

The noticeably white patch in the goal mouth, allowing for the inherent randomness in the sampling of the dataset, is likely due to the tendency of any player receiving the ball in this area to shoot on site. Very rarely does a player in possession of the ball in this area choose to pass (i.e., to continue the possession). Unfortunately, this action will mean they will not feature in this scatter plot due to the exclusion of shot events.

5.7.2 Passing Style and Chances



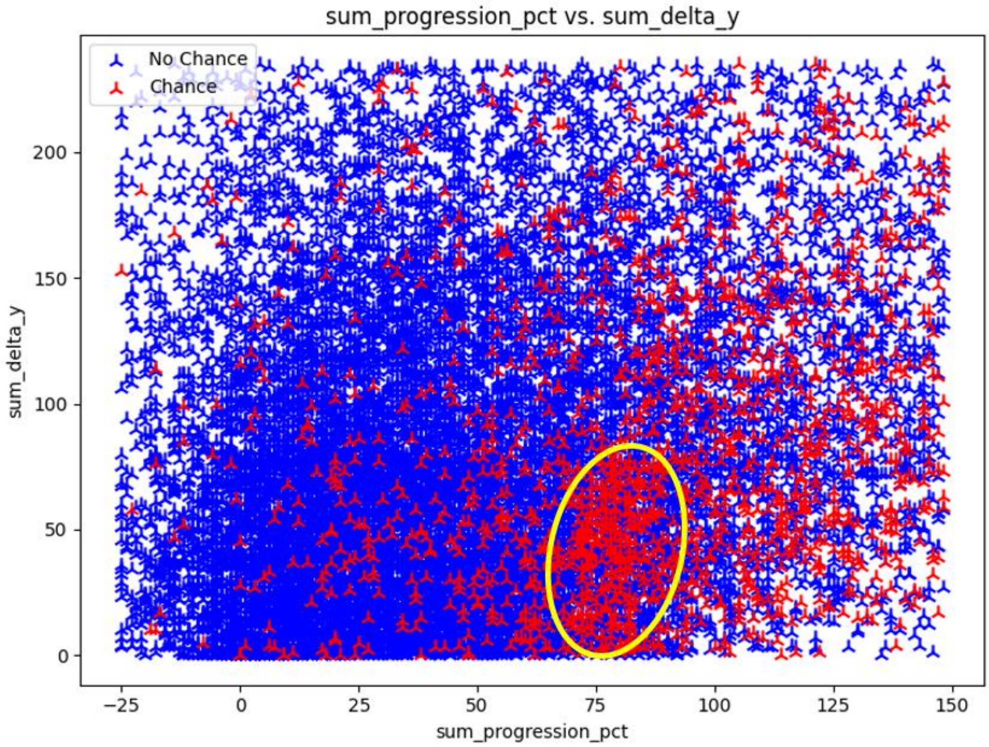
Football tradition dictates that teams can be separated into two groups; those that prefer to "keep it on the deck", passing the ball accurately and repetitively over short distances along the ground (a.k.a. the technical "tiki-taka" teams), and those that like to "hoof it", launching the ball as quickly and as far up the pitch as possible (Burnley FC style[18]). As ever, the truth lies somewhere in midfield.

The above plot of 20,000 possessions suggests that most of the teams represented in the dataset prefer the shorter pass, which might be expected given the high standard of teams featured. Players at the highest levels are generally more technically adept and will typically try to maintain control using short passes in their possessions. This is visualised as the denser, wider spread of possessions denoted in either blue or red towards the bottom of the scatter plot. Additionally, the possessions that result in Chances, in red, could reasonably be described as clustering together towards the bottom-right of the plot (ringed), which suggests that, generally, shorter passing is more likely to successfully move the ball down the pitch, increase progression percentage, and create Chances.

Teams judiciously using longer passes are not completely without hope, however. It could reasonably be suggested that there is another valid, though less dense, cluster of Chances (top centre, ringed) where the average pass length is greater, yet teams have still made significant progress towards the goal during those possessions. This demonstrates that the longer pass, if deployed accurately, can potentially lead to Chance creation, though at a somewhat less reliable frequency.

[18] https://www.premierleague.com/stats/top/clubs/total_long_balls?se=363

5.7.3 Crossing and Chances



Plotting the total change in Y co-ordinate versus the sum progression percentage of 20,000 possessions reveals a cluster highlighted in the yellow ring above. This clustering is likely to be explained by the action of crossing the ball, either from open play or from corner kicks. 'Crossing' is a synonym for passing from wide areas of the pitch to central areas, closer to the goal. Crossing towards the centre of the pitch near the goal will cause the distance to goal to reduce by approximately 60-90%, with variation depending on the precise angle of the pass (see below). This red cluster demonstrates the importance of this pattern of play in Chance creation.

The table below shows the variations in progression percentage for different cross start and end locations that account for this phenomenon. Corner kicks are taken, for example, from [120, 0].

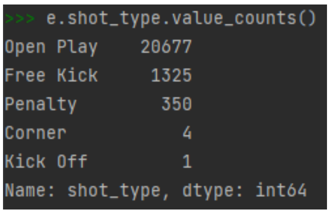
Cross Location Points		Distances		Progression %
Start	End	To Goal Start	To Goal End	
[90, 2]	[114, 40]	48.4	6	88
[95, 5]	[114, 40]	43.0	6	86
[105, 15]	[114, 40]	29.2	6	79
[110, 20]	[114, 40]	22.4	6	73
[110, 10]	[108, 40]	31.6	12	62
[120, 0]	[114, 40]	40.0	6	85

6. Assumptions

Several assumptions about the data and the approach towards processing and modelling have been made, as described here.

- **As the crow flies.** Distances and lengths are approximated by assuming straight lines between start and end locations for any event type. Essentially, Euclidean distances are used in all calculations.

- **"No Penalty!"**. Shots taken directly from "dead-ball" situations such as penalties and free-kicks are excluded from representing a typical Chance as there is no play to model before a shot is taken. They account for a minority of Chances in the dataset as this summary count of event "shot_type" attributes demonstrates.



```
>>> e.shot_type.value_counts()
Open Play     20677
Free Kick      1325
Penalty         350
Corner            4
Kick Off          1
Name: shot_type, dtype: int64
```

  - In fact, penalties are interesting enough to merit their own study, especially given the events of Euro 2020 this summer.[19]

- **A shot is a shot**. There is a direct correspondence between a Statsbomb event of type "Shot" and a Chance. That is, any event of type "Shot" will be labelled as a positive example of the target Chance class.

- **All games are equal**. All matches in the Statsbomb dataset are representative of general play in football matches. The dataset contains match event data from different competitions – national leagues, European competitions, international competitions and both men's and women's games. The assumption here is that a consistent analytical approach can be applied to this dataset

- **Timing**. No allowances will be made for ***when*** sequences take place e.g., whether they occur in the first minute of the match or in the final half of extra time. Any degradation of performance that may be attributed to time/stamina (within a game or across the season) will be disregarded.

- "**They're unstoppable**". As the attacking team mounts the offensive, the opposition will naturally defend. The models employed will not explicitly factor in any defensive action. This is because the defensive aspect of the game is not the focus of the project.

- **DIY xG**. Expected Goals (xG) is a value that is typically provided only for 'shot' events. Statsbomb, the data source used, will only provide xG values for explicit attempts on goal.

[19] https://www.theguardian.com/football/2021/jul/11/italy-crush-england-dreams-after-winning-euro-2020-on-penalties

Given that there is no available source of xG values for every possession sequence, only ‘shot’ events, I have implemented my own xG model. The assumption is that the use of a home-made xG for all possession sequences is better than sourcing xG from diverse providers as a composite.

7. Methodologies

"You can plan, but what happens on a football field cannot be predicted."

Manuel Neuer

This section describes the methodologies used in applying the machine learning toolkit to the suitably prepared dataset.

Match event sequences can be identified and labelled as those that lead to an attempt on goal (a Chance) and those that do not. Machine Learning models can be trained under supervision to distinguish between the two labels; hence the problem becomes one of binary classification. A trained model could then be used to determine whether an unseen event sequence will result in a Chance.

Each match event sequence may be assigned a calculated expected goal (xG) value. This value depends on the location of the final event in the sequence. The xG value is a measure of the quality of Chance. The task can then be reframed; not as a classification of Chance, but as a regression of the quality of Chance on the match event sequences. As suggested in the project proposal feedback, a trained model can then be used to predict an xG distribution for unseen match event sequences

7.1 Logistic Regression applied to Sequence Classification

To enable a logistic regression model to fit the event data, the variable length sequences of event data must be aggregated or summarised in some way. The predictor event data X must be formed into a single input row corresponding to each Y Boolean response variable that describes whether a Chance was created or not.

Focusing on progressive action events, the aggregations should summarise characteristics of the sequence of passes or carries that occurred in each possession.

The pass and carry features were therefore aggregated using the following methods:

- Pass length total sum and variance
- Carry length total sum and variance
- Pass speed mean and variance
- Carry speed mean and variance

These values are calculated for each possession using the convenient, built-in aggregation functionality provided by a Pandas DataFrame. The aggregated values are assembled into another DataFrame along with the corresponding Boolean response label, signifying whether each possession results in a Chance. A logistic regression model can then easily be fitted using the Python Scikit-learn library.

7.2 The ABC of xG

Expected Goals (xG) is a useful metric for comparing the relative quality of Chances. The xG models the probability that a shot will result in a goal. The main driver in the xG model is the pitch location of the shot, with more complex, proprietary xG models including many additional factors.

Statsbomb have implemented their own proprietary xG model but only attach that xG value to events of type “Shot” (i.e., Chance) in the dataset. While this is useful for possession sequences that result in a Chance it unfortunately means that any possession that does not result in a “Shot” is lacking a comparative xG value.

If there was an xG value available for every event, or at least every end of possession, it would be possible to train a model to learn the pattern between event data sequences and output xG. To generate such a dataset of xG values, I decided to implement my own basic xG model to fill in the missing values.

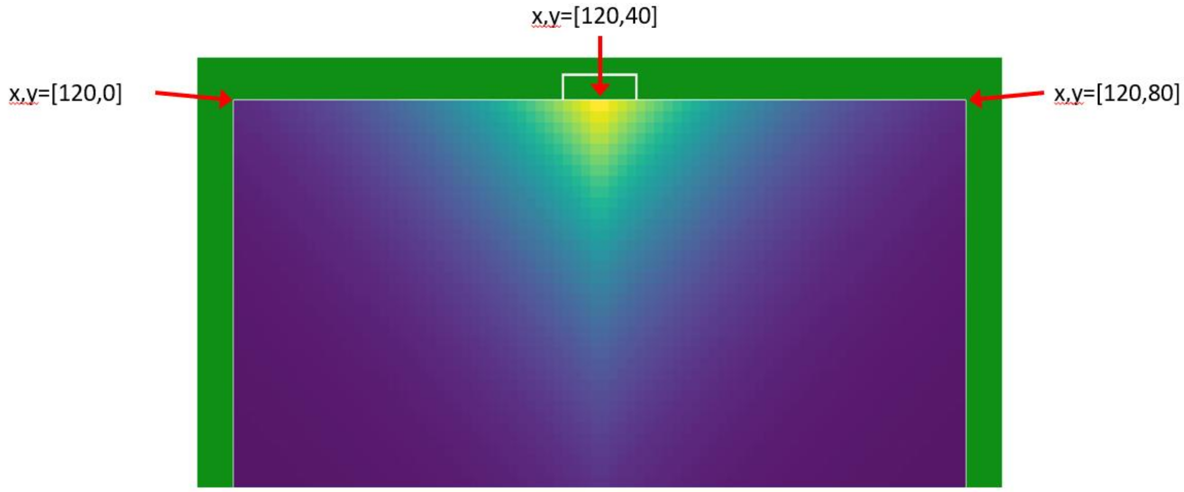
The xG model implemented also uses logistic regression to model the probability of a goal being scored. For every event of type “Shot” in the dataset, I distinguish between those that result in a goal (using the “shot_outcome” attribute) and those that do not and assemble the pitch coordinates of each of these shots. Once this data is assembled – over 20,000 shots - a logistic regression model can be fitted with response variable Y (Boolean goal scored) against predictor variables X (pitch coordinates). Every location on the pitch (0-120, 0-80) can be input to the model to output an associated probability for that location.

I, consciously, made a simplifying assumption that there was symmetry to the xG distribution across the width of the pitch. As such I decided to flip the coordinates of the “right sided” shots to the left so that the logistic regression could more easily linearly separate the two classes and learn the difference between a goal and a non-goal.

I am aware that there are other variables at play which would undermine the outcome mirroring I used; for example, right/left dexterity or perhaps favoured team attack approach. However, I made a judgement that the resultant DIY xG was fit for purpose.

I deliberately excluded penalties from the dataset analysed for the xG analysis as these have a detrimental impact on the result set, skewing the probability distribution of goal scoring based on pitch location.

Below is the predicted xG output by this model for each integer coordinate location on the pitch, plotted as my own xG heatmap:



Models used for comparison are described in the footnote references [20] and [21].

The diagram would indicate that there is a higher probability of a goal when an event sequence concludes in centre of the pitch, and towards the goal. The xG increases as X tends towards 120, and Y towards 40.

The DIY xG model suggests the goal probability is around 40 percent in the yellow zone and this does indeed closely resemble the established xG models referenced. This emphasises the criticality of pitch location in the determination of Chance creation and goal-scoring.

After this operation, every possession can be assigned a corresponding concluding xG value. This dataset can then be used to train a model to perform a regression, and then predict an xG value on unseen test event sequences.

7.3 Variable Length Sequence Modelling

Each possession can be represented as a matrix of real values of width N and height M, where N is the number of event features selected and M is the number of events in the possession sequence. Each of these variable height matrices can then be labelled accordingly as resulting in a Chance (or not), or with an associated xG value: that is, the xG associated with final event in the sequence.

A recurrent neural network can be trained with a collection of these labelled variable length patterns. This can be achieved by using a mask to pad the variable sized matrices to a consistent size, whereby each matrix will be the height of the longest possession sequence. Any sequence shorter than the longest sequence will be appended a specifically defined padding value to fill in the gaps, which the network is then directed to ignore.

A network configured in this way can then learn to classify possession sequences according to the Chance labels and assign a probability that each possession belongs to either class. To measure the

[20] https://www.fantasyfootballfix.com/blog-index/how-we-calculate-expected-goals-xg

[21] https://statsbomb.com/2016/04/explaining-and-training-shot-quality

classification error at each training step, the network applies a binary cross-entropy loss function, and then back propagates errors to adjust the network weights to repeatedly minimise the loss. After training, a test set is used to predict a class label for each test possession, with the output being an associated class probability; the probability that the test possession belongs to the Chance class. This probability can be assumed to be the chance of a Chance.

Similarly, a network can be trained to perform a regression of an associated xG value on the variable length patterns, again using a mask to pad the matrices to a consistent size. To perform the regression, a different loss function must be used - in this case, mean squared error. Again, the errors are back propagated to adjust the weights to minimise the loss during training. After training, the test set is used to predict an xG value for each test possession.

7.4 Text Classification & Sentiment Analysis

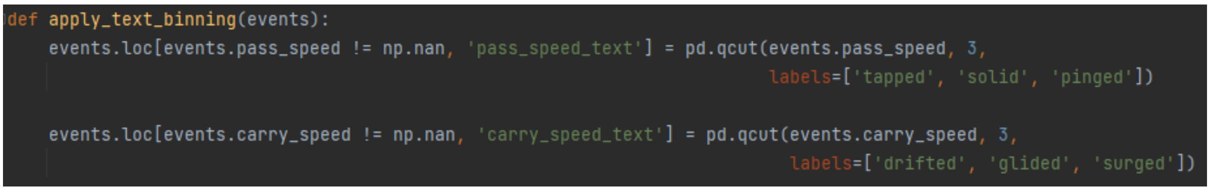
Natural Language Processing seeks to model sequences of elements (typically words) and the inherent patterns and dependencies between them. Considering the nature of the data being analysed, NLP would not suggest itself as the best approach. However, it occurred to me that NLP could be a powerful tool in Chance prediction if I were to represent the numeric sequences of football match event data as text and then task a neural network with learning how to classify those same text sequences. Thus, the model could learn the language of football.

To enable this approach, the matrices of real values that describe the event features of each possession must be transformed into representative and descriptive text. This can be achieved by using a technique called "binning" (or sometimes, "bucketing").

Each real value is assigned one of a small number of interval labels according to its magnitude. This method can be applied to each of the pass and carry features that are present in the matrices of match event sequences. Investigating methods for allocating labels in this way to generate the text could be the subject for a separate project. Here, as this is more of an initial proof of concept, the labels are assigned evenly into three bins, based on the distribution tertiles of values for each feature. Put another way, after the text labels have been assigned, there are approximately an equal number of events in each labelled bin. An obvious future avenue for exploration would be to increase the number of bins, and therefore the vocabulary, used to describe the events. The table below documents each label.

Feature	Low Label	Medium Label	High Label
pass_speed	'tapped'	'solid'	'pinged'
pass_length	'short'	'midrange'	'longball'
carry_speed	'drifted'	'glided'	'surged'
carry_length	'step'	'advance'	'keptgoing'

Below is an example code snippet that demonstrates use of the Pandas “qcut” method to facilitate this operation at scale on a DataFrame:

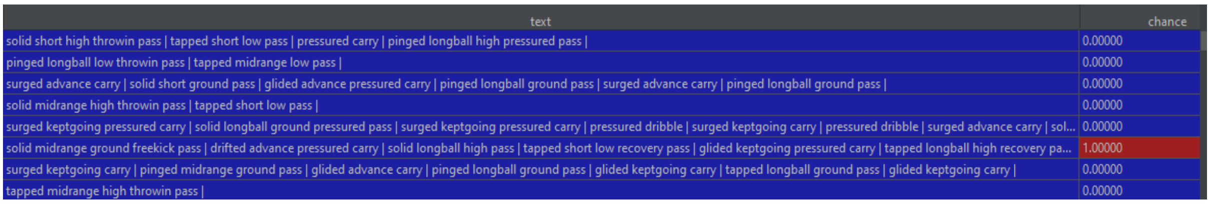


```
def apply_text_binning(events):
    events.loc[events.pass_speed != np.nan, 'pass_speed_text'] = pd.qcut(events.pass_speed, 3,
                                                                          labels=['tapped', 'solid', 'pinged'])

    events.loc[events.carry_speed != np.nan, 'carry_speed_text'] = pd.qcut(events.carry_speed, 3,
                                                                            labels=['drifted', 'glided', 'surged'])
```

After the numeric features have been binned using this method, the text sequences are assembled by aggregating the individual text descriptions of each event within each possession. This process generates a single text string that describes each possession. An event level delimiter is included for clarity.

The next step is the application of an appropriate binary labelling indicating Chance or No Chance. This positive or negative nature can be analysed through Sentiment Analysis of these “commentary documents”, a few examples of which are shown below.



text	chance
solid short high throwin pass | tapped short low pass | pressured carry | pinged longball high pressured pass |	0.00000
pinged longball low throwin pass | tapped midrange low pass |	0.00000
surged advance carry | solid short ground pass | glided advance pressured carry | pinged longball ground pass | surged advance carry | pinged longball ground pass |	0.00000
solid midrange high throwin pass | tapped short low pass |	0.00000
surged keptgoing pressured carry | solid longball ground pressured pass | surged keptgoing pressured carry | pressured dribble | surged keptgoing carry | pressured dribble | surged advance carry | sol...	0.00000
solid midrange ground freekick pass | drifted advance pressured carry | solid longball high pass | tapped short low recovery pass | glided keptgoing pressured carry | tapped longball high recovery pa...	1.00000
surged keptgoing carry | pinged midrange ground pass | glided advance carry | pinged longball ground pass | glided keptgoing carry | tapped longball ground pass | glided keptgoing carry |	0.00000
tapped midrange high throwin pass |	0.00000

The analysis of Chance creation is thus reframed to be very similar to a traditional sentiment analysis problem such as the classic IMDB movie review rating prediction. Review ‘documents’ for each movie are provided alongside an indication whether the review for the film is positive or negative. The model will, through sentiment analysis, learn how to evaluate whether the text in the documents provided would imply a positive or negative review.

Taking the analogy further, if movie reviews can be classified as positive or negative, then they can also be graded, for example, out of five stars, or even on a ‘Tomatometer’[22].

Similarly, if commentary documents can be classified as creating a Chance or not, then they too can be graded, in this case by some measure of Chance quality i.e., the xG value. By assigning xG values to each commentary document and training a model to perform a regression of xG on the commentary, predictions can be made about unseen test commentary documents.

[22] https://www.rottentomatoes.com

8. Implementation

8.1 Python

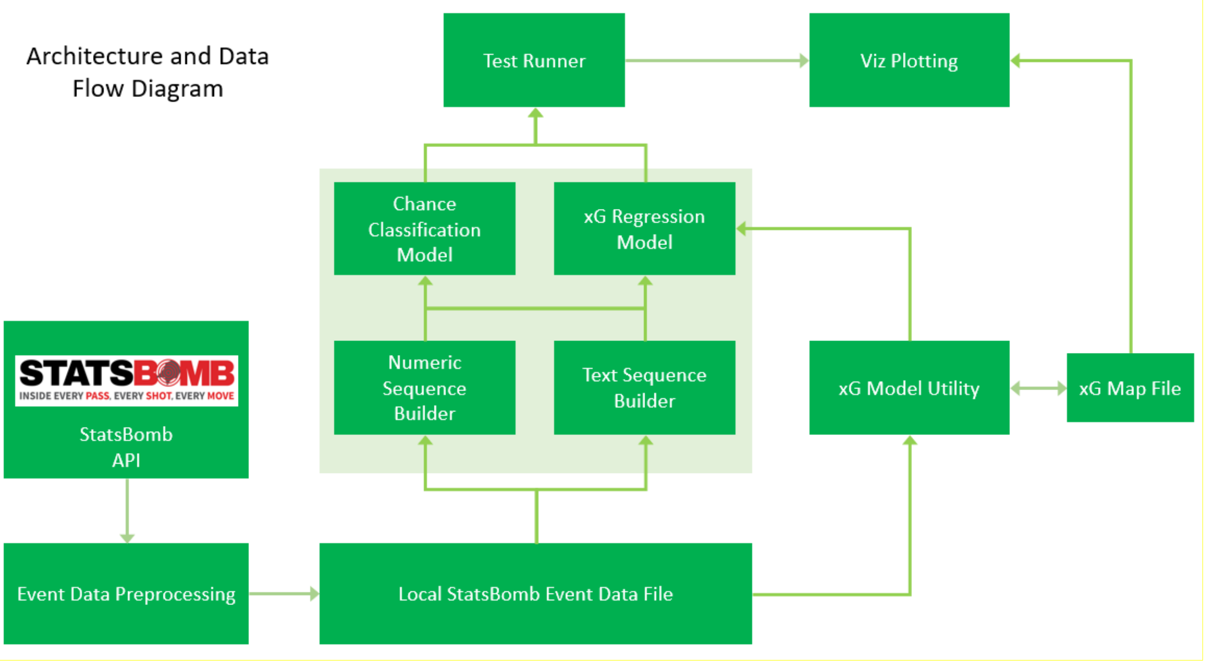
The Python programming language is used for the implementation. Python is widely recognised (and taught) as a first-class choice for implementing solutions for data science and machine learning problems. Additionally, I have some experience using Python during the modules of the MSc course and prior work experience.

Using Python grants me access to many helpful third-party libraries that implement machine learning models and neural network operations. This implementation makes heavy use of the Pandas library, in particular DataFrame objects for storing, manipulating, enriching and passing data between sources and analytic libraries. Many operations required to manipulate the data into a suitable format for processing are based upon the Numpy utility library and its associated data types.

The analytics to support the neural network modelling is largely built upon Tensorflow2.0 and the Keras API. Supplementary analytics such as the Logistic Regression implementation and various metrics utilities are provided by the Scikit-learn library. Matplotlib and Mplsoccer have been used for visualisations.

8.2 Architecture/Overview/Data Flow

The code is organised into the following logical, modular structure.



The interface to the remote StatsBomb data store (in Github) is called to extract the data as previously described, with some pre-processing and aggregation performed before the entire available dataset is stored locally.

The xG Model Utility makes use of the stored StatsBomb event data to assemble its distribution of shots and perform the logistic regression to build the xG model. The model is used to build a cache of predicted xG values for each grid coordinate on the pitch map.

The Numeric Sequence Builder randomly samples the stored possessions and formats the matrices of numeric event sequence data in preparation for training both the Chance classification and xG regression models.

The Text Sequence Builder randomly samples the stored possessions and generates the commentary documents in preparation for training both the Chance classification and xG regression models.

Model training and testing is automated by the Test Runner module, which can perform several trial runs depending on configuration. Categorised results of each run, model test predictions and training history are returned as the output to enable average performance metrics to be calculated if required.

The Viz Module provides some visualisation utilities for plotting training history, event feature scatter plots and xG heat maps.

8.3 References to Tooling Used

Pandas (https://pandas.pydata.org/)

Numpy (https://numpy.org/)

Statsbombpy (https://github.com/statsbomb/statsbombpy)

Tensorflow (https://www.tensorflow.org/)

Keras (https://keras.io/)

Scikit-Learn (https://scikit-learn.org/stable/)

Matplotlib (https://matplotlib.org/)

Mplsoccer (https://mplsoccer.readthedocs.io/en/latest/index.html#)

8.4 Neural Network Configuration

The choices of network hyper-parameters must be made to strike an acceptable balance between training efficiency (for given compute power) and modelling performance (accuracy/predictive power) of the trained network. Tensorflow/Keras API enables these parameters to be tweaked easily. A discussion of some key parameter choices follows:

- The number of layers of LSTM cells can be specified in the structure of the model. However, the universal approximation theorem states that a neural network with a minimum of one hidden layer between the input and the output can model any given function. It is therefore reasonable to configure models with a single hidden layer of LSTM cells for this project.[23]

- The number of LSTM cells within the hidden layer is also crucial. More cells will enable the network to model the target function more accurately at the cost of increased compute time. To decide on a suitable configuration, I performed simple indicative tests by training a series of networks with increasing numbers of LSTM cells on a reduced dataset over 50 epochs. The results (displayed below) indicate that a reasonable choice to balance predictive performance and efficiency is 64 or 128 cells.

Number of Cells	Binary Accuracy	Precision	Recall	time/step
32	0.7433	0.777	0.6674	10ms
64	0.8505	0.7895	0.9474	8ms
128	0.8794	0.8858	0.8653	12ms
256	0.9052	0.8514	0.9768	1s 37ms
512	0.9052	0.8676	0.9516	6s 189ms

- Dropout is used to regularise networks by excluding different random selections of connections from adjustment during each training epoch. This helps to reduce the potential for overfitting, as it is likely to break any recurring dependencies between network cells over the training phase and forces other connections to be explored. In addition, it has the added benefit of reducing the number of adjustments taking place within the network and therefore improves efficiency. I tested a range of dropout levels as shown below, ultimately settling on a reasonable value of 0.2 (though, it should be stated that the process is somewhat random).

Dropout	Test Accuracy
0.1	0.8794
0.2	0.8814
0.3	0.8464
0.5	0.8526

- Epochs specify the number of iterations during the training phase. Throughout the project I have varied the number of epochs depending on the situation. For example, while debugging I have typically used ten or twenty. For testing other hyperparameters I have increased this to fifty. For measuring test results, I have used 500 to 1000. The number used is specified with results.

[23] http://neuralnetworksanddeeplearning.com/chap4.html

8.5 Source Control

Throughout the development and testing phases I have employed Github as a repository and source control mechanism. I have performed regular commits to the repository. This has enabled me to maintain a history of the code base, keep a back-up of the code, and work across different machines as required. The code developed can be accessed via the following repositories for inspection.

https://github.com/Birkbeck/msc-data-science-project-2020_21---files-ehine02

https://github.com/ehine02/chance

8.6 Excel

An honourable mention goes to Microsoft Excel, which was initially useful for viewing the raw data once it was downloaded from Statsbomb. It was particularly helpful to apply filters to event types or to select individual possessions to make sense of how the dataset was arranged.

8.7 Testing

In addition to the usual debugging and integration testing required to construct modular software for complex tasks like this, I have implemented basic unit tests using the Pytest module for some of the more straightforward yet fundamental utility functions.

9. Results & Analysis

"The ball is round, the game lasts ninety minutes, and everything else is just theory."

Josef "Sepp" Herberger

At the start of this project, I had a somewhat romanticised theory that my widely shared passion for football could be attributed to the simplicity, synchronicity, and speed of movements by players as they create goal scoring opportunities. The results I obtained through analysis of the football match event sequence data only bolster my theory but do however prove that this data can be used to predict Chance creation to a high degree of accuracy (between 90% and 93%).

To successfully demonstrate that event data could be used to model the beautiful game, I imagined the dataset that I would need would be vast with multiple columns. I anticipated a gruelling iterative process of trial and error to fit the model and to prove the predictability of chance creation based on event sequences. As detailed, I spent a lot of time in ensuring I had the right data to prevent leakage and to truly assess the influence of the fundamental data attributes on Chance probability calculation. As it happens the vast dataset was not essential. For convenience I reiterate *The Fundamentals*; speed of pass, speed of carry, length of pass, length of carry. The results I obtained from the application of the model to simply these Fundamentals exceeded all expectation.

9.1 The Fundamentals – Pass & Carry

Having excluded location-related data in the dataset analysed, the results of the modelling indicate that the attributes considered to be the 'Fundamentals', namely pass and carry speeds and distances, are indeed good predictors of Chance creation. As evidenced in the table above, either modelling approach (numeric or text based) when trained over enough epochs can achieve in the region of 90% binary accuracy in predicting Chance creation on unseen match event sequences constructed from these four fundamental attributes. The outcomes confirm my hypothesis that, although pitch location is a key factor in Chance creation, other factors relating to the movement and interactions that take place on the pitch can be useful predictors of Chance creation.

9.2 Chance Classification Metrics

9.2.1 Baseline Logistic Regression

The baseline performance of the logistic regression performed using aggregated event sequence data was between 65 and 71% binary accuracy, depending on sample size, in predicting Chance creation.

Sample Size	Binary Accuracy	ROC AUC
5000	0.65	0.7
2000	0.64	0.71
1000	0.71	0.77
500	0.7	0.77

This variation can be explained as the increasing sample size increases the number of examples that cannot be separated using the simple linear model. In the context of binary classification, a completely dumb model would theoretically achieve 50% accuracy by predicting a Chance having been created for every possession sequence. Therefore, the Logistic Regression can be said to have more predictive power than this simplistic approach and serves as a useful benchmark for comparison to the more complex neural network models. However, as it is based upon a linear model and operating on aggregated event data, logistic regression is unable to model the non-linearity present in the match event dataset.

In the numerical sequence approach, the LSTM network trained over 1,000 epochs was able to achieve around 93% Chance prediction accuracy, with a recall of 97% and precision of 90%.

In the text sequence approach, the LSTM network trained over 1,000 epochs was able to achieve around 90% Chance prediction accuracy, with a recall of 95% and precision of 86%.

The results suggest that either approach is significantly more performant than the baseline logistic regression, and that the models demonstrate good predictive ability in classifying the possessions that result in a Chance.

9.2.2 Numerical Possession Sequences

Average Benchmark Performance Trials - models trained over 200 epochs

	Trial number			Average
	1	2	3	
Binary Accuracy	90.2	91.7	92.9	91.6
True Positive	723	774	765	754.0
True Negative	868	854	884	868.7
False Positive	157	136	107	133.3
False Negative	16	12	19	15.7

Results Table 1 – Classification metrics for Chance predictions made by a numeric sequence model trained for 200 epochs averaged over 3 trial runs

Longer Training Performance Trial – one-off 1,000 epochs

Overall binary accuracy 93.5%

Chance Created?	Actual True	Actual False
Predicted True	322	36
Predicted False	10	338

Results Table 2 - Classification metrics for Chance predictions made by a numeric sequence model trained for 1,000 epochs.

These results are encouraging, as the model demonstrates a high success rate in correctly classifying whether a possession results in Chance creation or not. The proportionally higher false positive rate compared to false negative (36 v 10) means the model is predicting more possession sequences that end in a Chance when they do not. In effect it seems this model is slightly optimistic.

9.2.3 Text Possession Sequences

Average Benchmark Performance Trials - models trained over 200 epochs

	Trial number			Average
	1	2	3	
Binary Accuracy	87.6	89.8	89.6	89.0
True Positive	737	737	729	734.3
True Negative	810	856	852	839.3
False Positive	151	137	163	150.3
False Negative	68	44	21	44.3

Results Table 3 – Classification metrics for Chance predictions made by a text sequence model trained for 200 epochs averaged over 3 trial runs

Longer Training Performance Trial – one-off 1,000 epochs

1,000 epochs – 90.6% binary accuracy

Chance Created?	Actual True	Actual False
Predicted True	299	49
Predicted False	17	339

Results Table 4 - Classification metrics for Chance predictions made by a text sequence model trained for 1,000 epochs.

These results are similarly encouraging, again demonstrating that the model can correctly predict Chance creation (or not) with high accuracy. Where the model has made incorrect predictions, the higher false positive rate compared to false negative rate (49 v 17) means the model is again predicting more possession sequences that end in a chance when they do not.

In both the text and numeric sequence models, the proportionally higher false positive rates could potentially be explained by the presence of defensive actions by the opposition. Even when the coordination and movements of possession sequences are almost entirely successful there is no guarantee of a chance being created if the opposition are ultimately able to repel the attacking actions before the chance is created.

Overall, however, the ratios of correct Chance class predictions to incorrect are very acceptable and both models perform well.

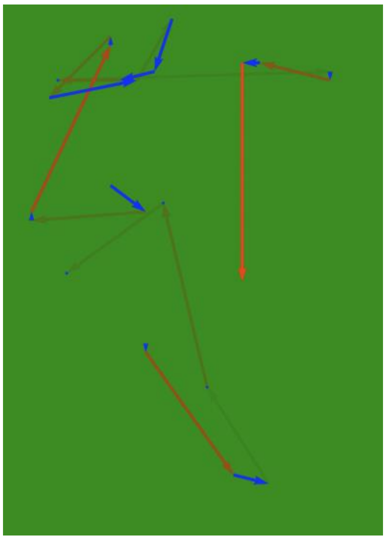
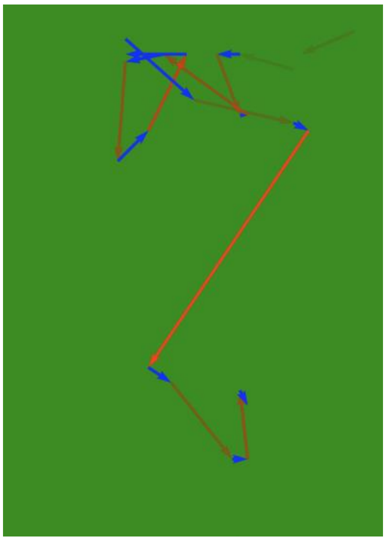
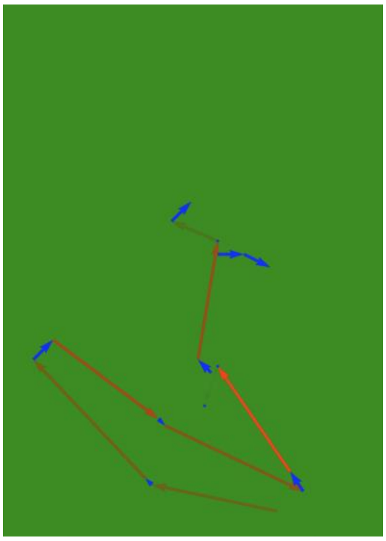
9.2.4 Exploring the Apparent Optimism of the Models

There appeared to be a prevalence of false positive predictions made by the models. That is, of the predictions the models got wrong, there were many more examples that were predicted to be Chances that in truth are not. To explore this issue further, some example possessions were selected, including false positive and true positive examples, and the sequences of passes and carries plotted on a pitch map for analysis.

The red arrows indicate passes, the blue arrows are carries. The alpha (transparency) of the arrows translates to the speed of the action (brighter = faster). Thus, some degree of expression of the features used as inputs to the model can be represented graphically.

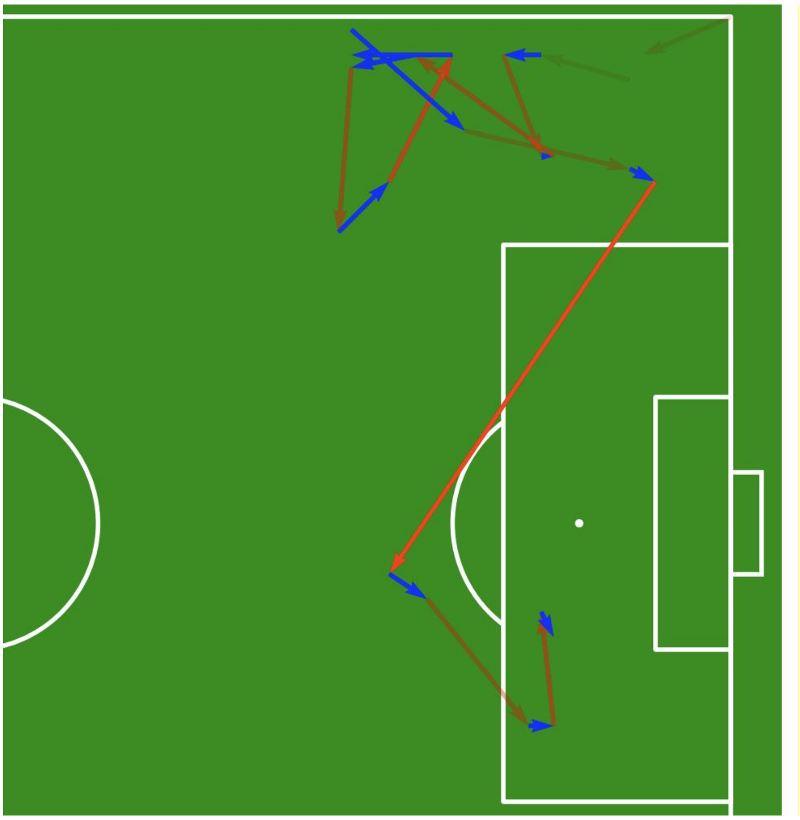
Spot the Chance!

Considering the three diagrams below, which one depicts the sequence of match events that result in a Chance?

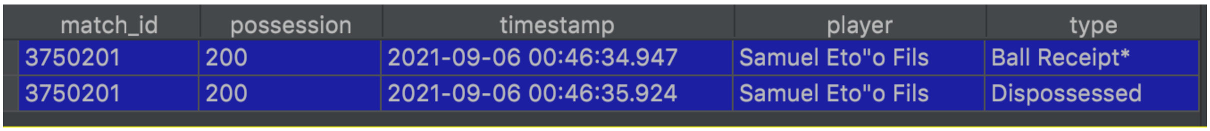
		

Let’s find out…

Example 1 - Possession 200 from match 3750201 (Barcelona v Manchester United). Depicted is an intricate Barcelona passing sequence consisting of 11 passes and 10 carries, starting from a corner kick. To describe the flow of this possession, one could say there is a series of short, tentative passes followed by a sudden longer, bolder (literally) switch to another more central area of the pitch, whereupon after another couple of slower passes, the sequence ends.

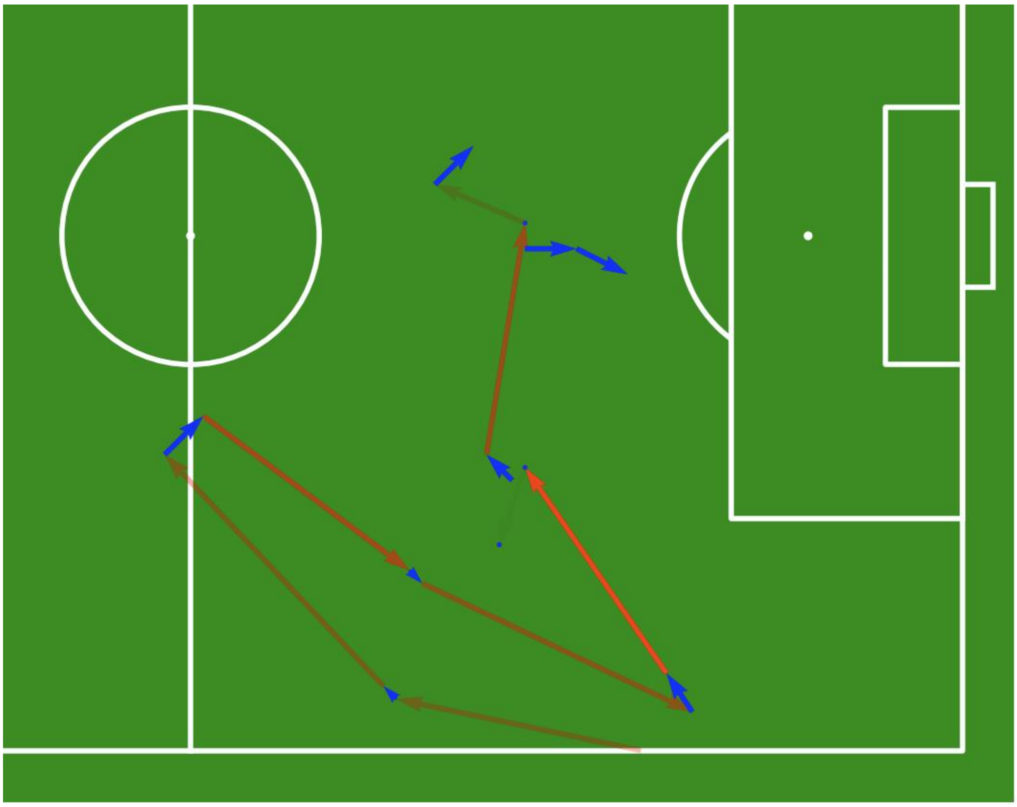


Although apparently promising, this sequence did not conclude with a Chance. Why not? Examining the raw event sequence reveals a defensive action resulted in a dispossession of the ball:

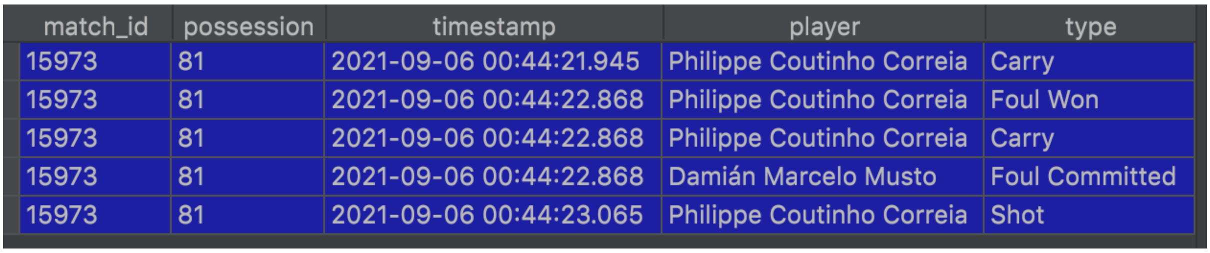


match_id	possession	timestamp	player	type
3750201	200	2021-09-06 00:46:34.947	Samuel Eto"o Fils	Ball Receipt*
3750201	200	2021-09-06 00:46:35.924	Samuel Eto"o Fils	Dispossessed

Example 2 - Possession 81 from match 15973 (Barcelona v Huesca) shares some characteristics with the previous example. From a throw-in, there is a series of more considered, slower passes (some very slow and therefore barely visible), and short carries. Then suddenly a quicker, bolder pass, followed by another couple of slower exchanges and movements.

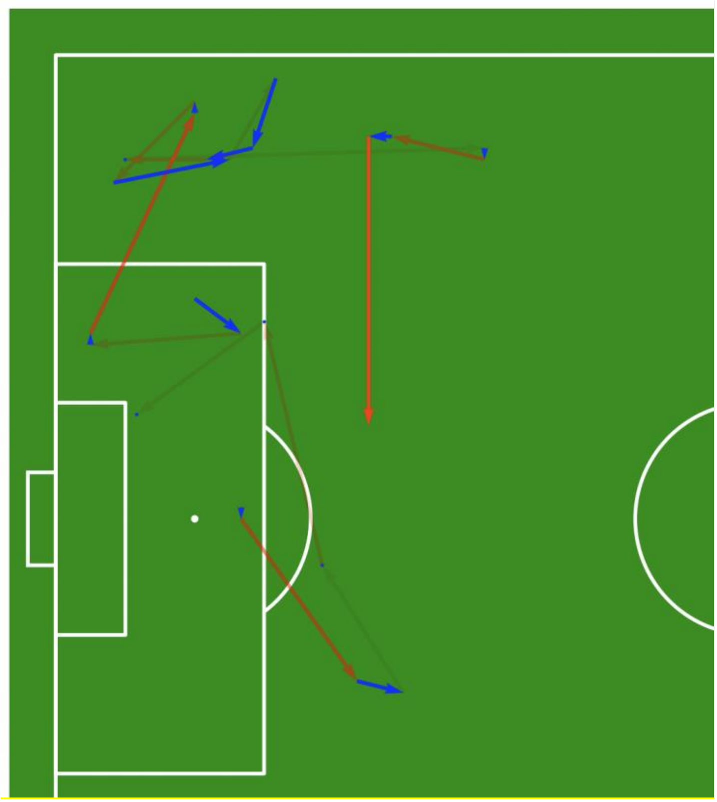


Success! This is the sequence that resulted in a Chance, i.e., a legitimate shot, despite the attacking player subsequently drawing a foul in the process.

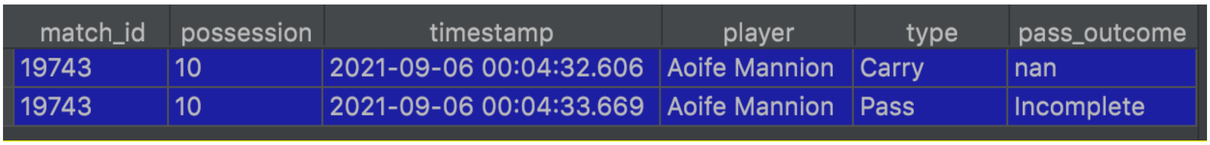


match_id	possession	timestamp	player	type
15973	81	2021-09-06 00:44:21.945	Philippe Coutinho Correia	Carry
15973	81	2021-09-06 00:44:22.868	Philippe Coutinho Correia	Foul Won
15973	81	2021-09-06 00:44:22.868	Philippe Coutinho Correia	Carry
15973	81	2021-09-06 00:44:22.868	Damián Marcelo Musto	Foul Committed
15973	81	2021-09-06 00:44:23.065	Philippe Coutinho Correia	Shot

Example 3 – Possession 10 from match 19743 (Birmingham City WFC v Chelsea FCW). Another example that could be described in similar terms to the previous two. Slower passes and carries connected by one faster pass and concluding with the fastest pass of the sequence. However, note the pitch location. This sequence is executed in the team's own half, close to their own goal. In fact, their goalkeeper is involved in the passing sequence. As discussed, the model is explicitly not allowed to know the specifics of the pitch location, therefore has no way of understanding that this sequence is very unlikely to produce a Chance. But the patterns of play as highlighted are like many of the sequences that do result in Chances.



Incidentally, this sequence ends in an incomplete final pass, and the opposition, Chelsea FCW, recover the ball.



match_id	possession	timestamp	player	type	pass_outcome
19743	10	2021-09-06 00:04:32.606	Aoife Mannion	Carry	nan
19743	10	2021-09-06 00:04:33.669	Aoife Mannion	Pass	Incomplete

These three examples demonstrate the difficulty in the process of learning to classify these sequences. The examples share similar characteristics, with each team successfully moving the ball around, varying the speed of play, and travelling towards the theoretically more dangerous area in the middle of the pitch, although this location is not explicitly described in the data as previously discussed.

However, the first example ends in a successful defensive action, the second in a Chance, and the third is taking place in the ‘wrong’ half of the pitch. This offers some explanation as to why the model tends to be optimistic in predicting Chances. There are more factors, of which the model is unaware, that can prevent the creation of Chances, even though the underlying patterns in the data suggest that a Chance should be created. Further analysis and work would be required to include some more of these missing factors in a model.

9.3 XG Regression Metrics

The following table presents regression metrics for xG value predictions made by numeric and text sequence models trained for 500 epochs averaged over 3 trial runs each, and one-off 1000 epoch trials

	Trials over 500 epochs			Average	1000 epochs One-off
R^2 score	1	2	3		
Numeric Model	0.67	0.71	0.81	0.73	0.83
Text Model	0.58	0.62	0.51	0.57	0.72

xG Regression Metrics

In the numerical sequence approach, the network was able to achieve an R^2 score of 83% after training for 1,000 epochs. This suggests that the numerical sequence model can explain and model 83% of the variability in the xG values solely by modelling the corresponding set of numerical event sequences.

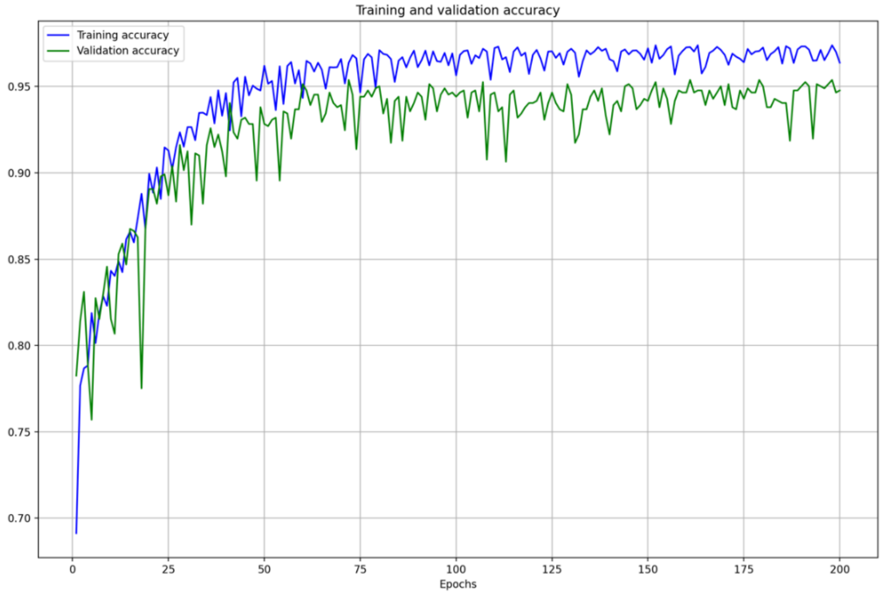
In the text sequence approach, the network was able to typically achieve a prediction R^2 score of 72% after training for 1,000 epochs. The model displays substantially more predictive power after training over a longer period. Extending the training period over even more epochs may improve performance even further.

Introducing the qualitative mapping aspect to produce the binning for the text commentary approach, at the expense of the quantitative numeric data, appears to reduce the precision of the text model when attempting to perform regression. In my opinion, this is to be expected. I would expect that binning features in this way would reduce their capacity for modelling a continuous output, as appears to be the case with the xG regression performed here.

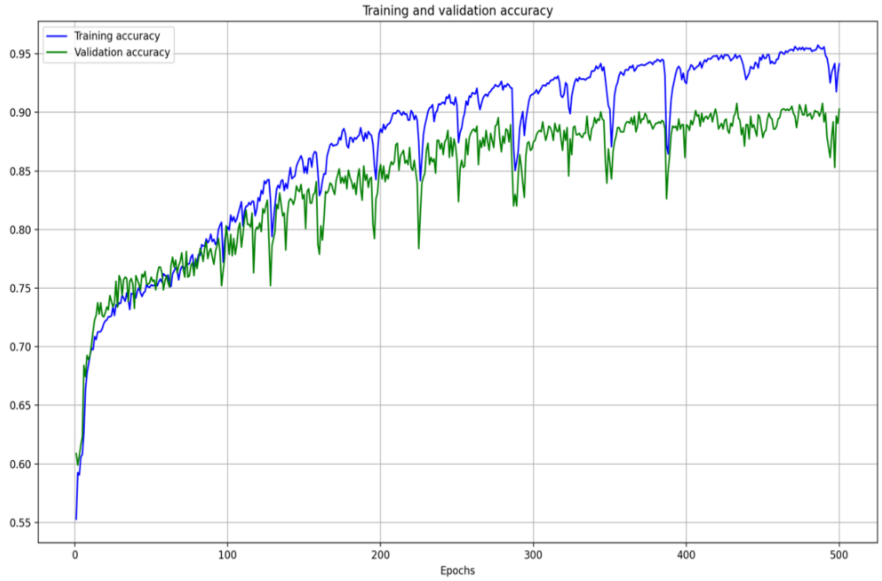
Nonetheless, the results suggest that both approaches have reasonable predictive ability, although the numeric sequence model has the edge. Both models can be said to reasonably predict the xG distribution for a given sequence of events within a possession but training the model using the numeric sequences results in appreciably improved predictive performance.

9.4 Example Model Training Visualisations

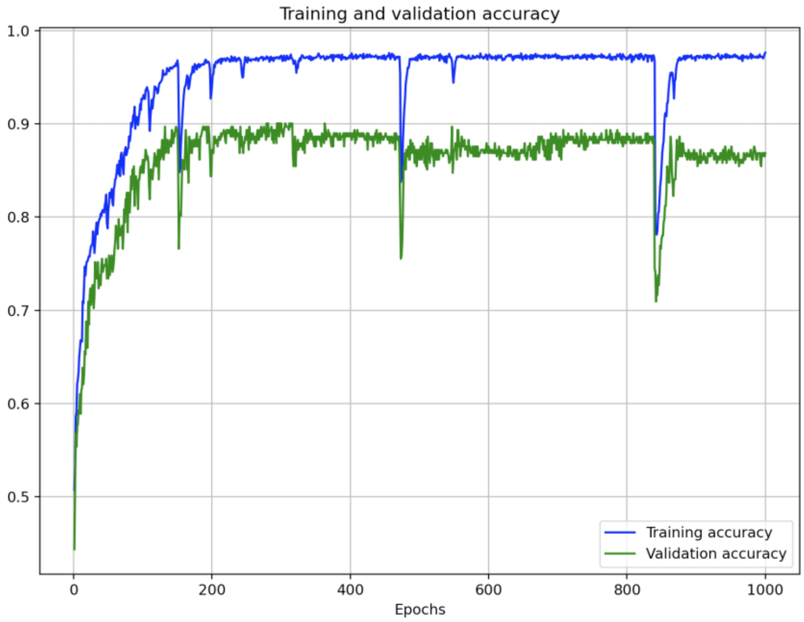
Example plot 1. Training and validation progress for the numeric sequence classification model over 200 epochs. The validation accuracy trails the training accuracy in this case, though it is stable and not noticeably diverging but shows more variability.



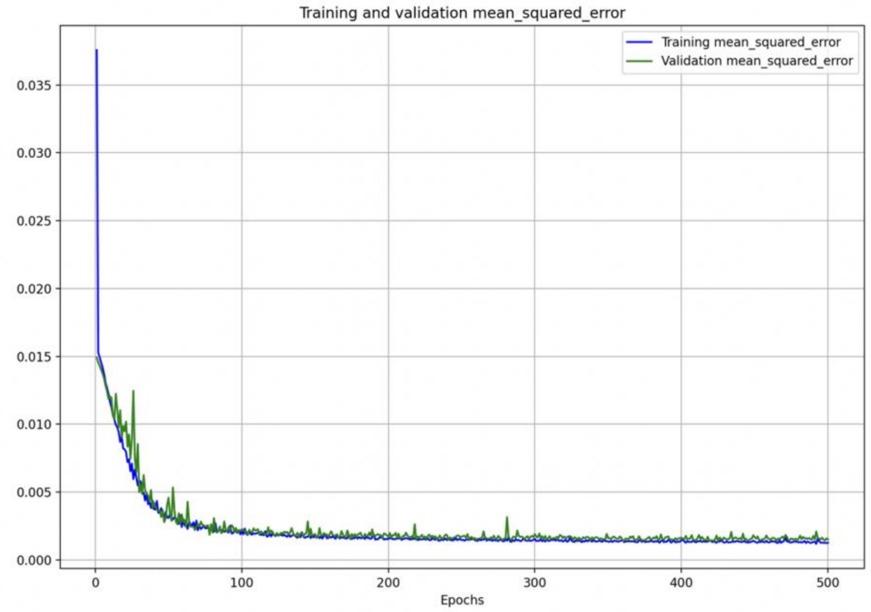
Example plot 2. Training and validation progress for the text sequence classification model over 500 epochs. In general, noticeable gains in training and validation accuracy are visible up to around 400 epochs.



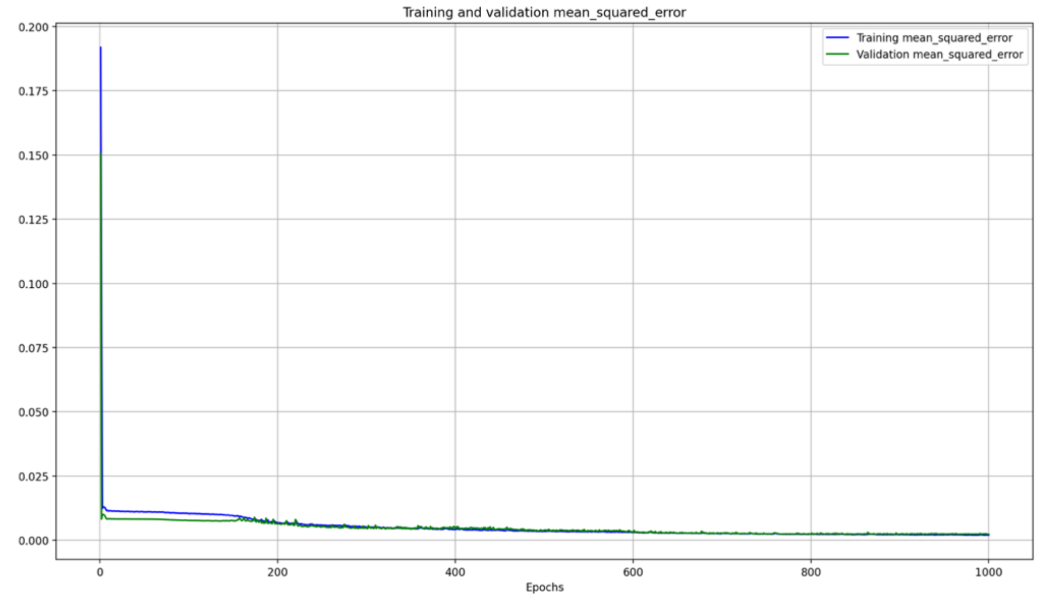
Example plot 3. Training and validation progress for the text sequence classification model over 1,000 epochs. It appears that validation accuracy is gradually diverging from the training accuracy. The model after training could arguably be said to be overfitting the training data.



Example plot 4. Training and validation progress for the numeric sequence regression model over 500 epochs. The number of epochs could be reduced as the model seems to have converged at 250-300 epochs.



Example plot 5. Training and validation progress for the text sequence regression model over 1,000 epochs. Although it is difficult to discern, slow and steady improvement appears to be made all the way up to 1,000 epochs.



9.4.1 To Locate or Not Locate?

As mentioned previously, I have consciously focused on non-location specific data, namely pass length, carry length, pass speed, and carry speed. These attributes are the fundamentals of the poetry of football, the movement between players, the precision of the pass and the creation of space. I do, nonetheless, acknowledge that location is a significant agent in the prediction of Chance creation.

I deliberately excluded location data to isolate the 'fundamentals' in the sequence analysis. However, I then chose to reintroduce location data to verify that this additional data simply enhanced the predictive power of the model. This constitutes a sanity check of sorts and effectively means a greater confidence in the original model results based only on the 'Fundamentals'.

This test could be performed with either the text sequences or the numeric sequence data. For the purposes of testing, I used the numeric sequence approach and, as the results are for indicative purposes only, trained the model over just **20 epochs**. The results are described below.

1. Baseline performance after only 20 epochs using 'Fundamentals' alone is 72%
2. Include Raw Location X & Y
    a. Tells the model exactly where the event is occurring on the pitch
    b. Accuracy versus 'Fundamentals' alone jumps to **86%**

3. Include Progressive move percentage.
    a. Tells the model how much nearer the ball is to goal after the effect of each event.
    b. Accuracy versus ‘Fundamentals’ alone increases to **89%**

Even over the limited 20-epoch training period, in each case the model learns to better classify the Chance creating possessions and makes more accurate predictions based upon the location information, which as mentioned is obviously a critical factor for chance creation. Therefore, I can be reasonably confident in the predictive power of the model using the movement attributes alone.

10. Conclusions

“The first 90 minutes of the match are the most important.”

Bobby Robson

Referring to the goals outlined in the objectives section, conclusions can now be drawn.

10.1 Speed & Precision

Chance creation is dictated by Speed and Precision in patterns of play

The results obtained suggest that the creation of Chances can indeed be predicted by considering just four measures of ball movement – namely pass length and speed and carry length and speed. There appears to be reasonably strong predictive power in models that are trained using these event features to forecast Chance creation. Therefore, it is reasonable to conclude that these four measures relating to match events are critical in the creation of goal scoring Chances.

10.2 Using RNNs

Recurrent neural networks can learn to classify the patterns of play that lead to Chance creation

The results obtained suggest that recurrent neural networks are indeed able to learn to classify the numeric representations of match event sequences that precede a scoring Chance. It can therefore be reasonably concluded that the movements involved in attacking the goal can be discerned independently from the location on the pitch.

10.3 Event sequences as text

Patterns of play can be transformed into descriptive text and Natural Language Processing models can learn to classify the texts that describe Chance creation

The project demonstrates that numeric data describing the fundamental movements of an attacking team’s possession can be reasonably encoded into descriptive text using binning. The results obtained suggest that once this process has been applied, Natural Language Processing models can be trained to classify the text describing the match events preceding a Chance. While the predictive power of models trained using this technique is strong, it is slightly lower than those models trained using the raw numeric data.

This small difference can likely be explained by the binning process, as fine details and granularity within the numeric data are lost. For example, passing speeds are mapped from a continuous scale to one of three buckets (‘tapped’, ‘solid’, ‘pinged’). This mapping will reduce the ability of a model to discern between different movements in the patterns of play. However, adopting the NLP approach still yields good accuracy of predictions on the test set.

10.4 Event sequence xG regressions

Expected Goal (xG) values can be predicted by a regression of the data describing the patterns of play

I was able to build a reasonable xG model, by using the location data of the shot events in the dataset. This xG model was used to assign an xG value to each possession sequence. An LSTM neural network was then trained to perform a regression of the xG values on both numeric and text representations of the match event sequences. The results show that the model resulting from this approach has reasonable ability in predicting xG levels for unseen test sequences. Again, similar small differences were observed between the text and numeric sequence approaches, and explainable using the same logic as above (binning).

10.5 The Final

"They think it's all over..."

Kenneth Wolstenholme

The introduction posed the question of whether data science could determine if there was meaning in the data describing a game of football. My conclusion is that this is possible. The results of the project suggest that analysis of the interactions (passing) and movements (carries) alone is sufficient in predicting Chance creation to a high degree. While the Recurrent Neural Network effectively learns and predicts Chance creation, perhaps your fellow fan, half-risen from their seat, taking a sharp intake of breath as they see the Chance unfolding, is subconsciously doing the same.

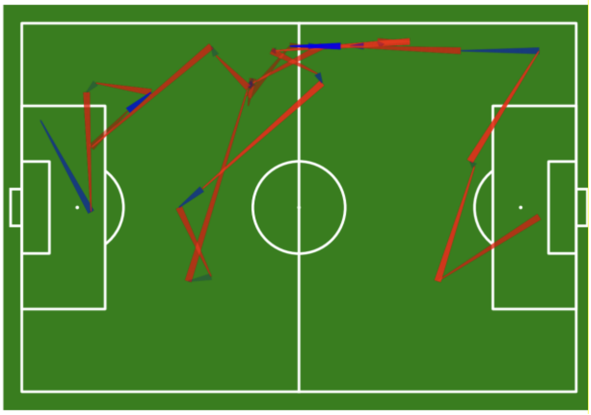
11. Areas for Future Investigation

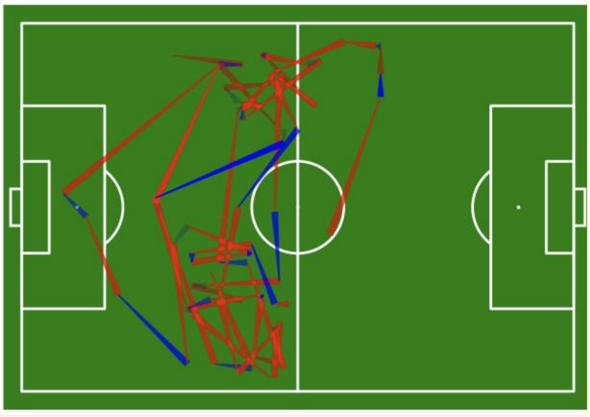
Potential avenues for further investigations have tempted me throughout the course of the project. The work required meant that inclusion of these additional explorations was simply unfeasible for the timelines of this project.

11.1 Image Classification Approach

I have explored two approaches to predicting Chance creation using RNNs, representing the match events as numeric sequences and text sequences. It would be interesting to consider the next step by treating Chance prediction as an image classification or character recognition problem.

I started to investigate this approach by plotting each event sequence on a pitch map using a colour codes for types of events (pass-red, carry-blue for example) and the alpha (transparency) of the colour representing the speed of each event. Below are a couple of example plots.





This frames the problem as a classic supervised image classification/character recognition problem. Instead of characters, the model learns to distinguish the patterns of event sequences that create Chances. This could be achieved using a convolutional neural network to learn the patterns, supervised by the corresponding Boolean Chance labels.

The potential pitfall here is that the location data is implicitly embedded in each image, resulting in the data leakage issue mentioned earlier in this analysis. This might be addressed by applying some sort of normalisation or offsetting to the coordinates involved, perhaps randomly flipping the images, or some other approach that attempts to “hide” the goal from the model.

11.2 Defensive Actions

To address a model’s apparent proclivity for false positive predictions i.e., predicting the creation of Chance when one does not arise, one could introduce factors to represent the influence of defensive action. This potential improvement could require information regarding the location of defender(s) or evaluating areas of pitch control. This would require further data sources, including tracking data for each individual player on the pitch.

11.3 Commentary

An interesting idea would be to make use of real match commentary from other sources to both train and test the model – that is, not the generated text as implemented by the dataset binning in this project. For example, transcriptions of match commentators, live match reports and so on. Does the language, writing or even volume and tone of voice indicate when chances are being created in passages of play?

11.4 Defining Patterns of Play

A game of football contains many common repeated ball movements and familiar patterns of play. Given the X,Y coordinates of the movements, it is possible to define when these particular patterns occur, and even to what extent. This information might prove to be a further useful indicator of Chance creation. For example, it is common for teams to "switch play" in possession when building an attack, moving the ball quickly from one side of the pitch to the other. This action could be measured by considering the delta in the Y coordinates of each event. Other patterns such as spreading play, one-twos, and lay-offs could be similarly defined and considered.

11.5 Allowing for variations in level of play

Segregating the data according to the team or competition level may reveal some interesting differences in the analysis. It would seem reasonable to expect teams that play at higher levels of the game to perform the progressive actions at higher speeds and with greater precision. Would Chance creation analysis hold for Grimsby Town v Bromley as it does for Liverpool v Real Madrid?

11.6 Incremental Chance Creation Analysis

The analysis in this project considers each possession event sequence in its entirety. By performing analysis incrementally on possession event sequences one event at a time as they build, it would be possible to model the change in probability of Chance creation throughout the evolution of each possession. It is likely that there are certain "key" events in each sequence that dramatically alter the ultimate probability of Chance creation resulting from the possession sequence. This approach may identify the presence of those key events.

12. Glossary of Terms

Pass – the action of transferring the ball between two teammates who are naturally in different locations on the pitch, labelled in the dataset with event type “Pass”.

Carry – the action of a single player moving the ball between two locations on the pitch, labelled in the dataset with event type “Carry”. Despite the suggestion in the term, the player typically uses their feet to *carry* the ball.

Possession – a sequence of 1..n events where a single team controls the movement of the ball by performing passes or carries. Possessions typically end in either a turnover (that is, the opposition team gains possession of the ball through an error or opposition action) or a Chance. Statsbomb provide a monotonically increasing integer ID to label each successive possession in a match

Chance – any attempt to score a goal by an attacking team after a sequence of events in a possession within a match. For the purposes of the project, and for simplification, all events in the StatsBomb dataset of type “Shot” are classified as a chance.

Expected Goals (xG) – a probability that describes how likely a shot is to become a goal. There are many different xG models, each including different factors to determine this probability. There is no standard definition of what factors should be included (which allowed me to select these when creating my own xG model).

Progression Percentage – the percentage change in distance from the ball to the goal at the start and the end of the corresponding event. If the ball has moved closer to the goal, then this will be a positive percentage. If the ball has moved further away from the goal, then this will be a negative percentage.

Numerical Possession Sequence or **Numerical Match Event Sequence** – a matrix of real (float) values of size n by m, describing the variations in m event attributes (e.g., pass speed, carry length) over the course of n events in a possession.

Text Possession Sequence or **Commentary Document** – a string of compact English words (or tokenized phrases) describing the level of variations in m event attributes (e.g., pass speed, carry length) over the course of n events in a possession.

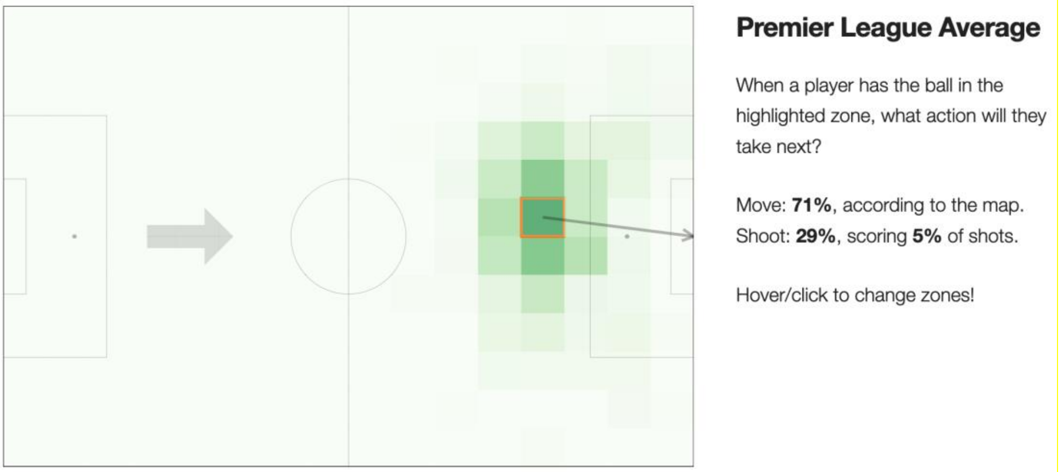
13. Literature Review

13.1 Expected Goals

Expected Goals (xG)[24] is a probability-based measure of scoring chance quality. Definitions are widely available, and models vary in complexity, but as a basic definition per Bundesliga.com[25], xG is "a predictive model used to assess every goal-scoring chance, and the likelihood of scoring.". xG is therefore primarily concerned with the outcome of a Chance, whereas this project's focus is the creation of those Chances. However, as xG is widely used as a measure of Chance quality, it is reasonable to investigate the regression of xG values on a selection of independent variables that constitute match event data.

13.2 Expected Threat

Singh introduces *Expected Threat (xT)*[26], a metric that attempts to quantify potential scoring probability based on ball possession and location in pre-defined pitch zones. As this method is possession based and concerned with scoring potential, it has more in common with this project's goals. The basis of this approach appears to be a Markov model, where each of the pitch zones equates to a state and each transition has an associated probability.



Premier League Average

When a player has the ball in the highlighted zone, what action will they take next?

Move: **71%**, according to the map.
Shoot: **29%**, scoring **5%** of shots.

Hover/click to change zones!

Figure 2 pitch zones in the Expected Threat model

13.3 Markov Model

Rudd[27] presented methods employing Markov chains to evaluate player level performance. Player actions are evaluated in terms of a transition matrix that measures the changes in goal probability per state transition. Players accumulate positive or negative changes as they execute different actions during the game. Players are then ranked according to these accumulated goal probabilities.

[24] https://en.wikipedia.org/wiki/Expected_goals

[25] Bundesliga.com

[26] https://karun.in/blog/expected-threat.html

[27] http://nessis.org/nessis11/rudd.pdf

13.4 Expected Possession Value

Expected Possession Value (EPV) is an ensemble model defined by Bornn et al.[28]. The model uses match optical tracking data describing each moment of a possession to compute the scoring probability. The output is a real value that instantaneously quantifies which team's players are optimally positioned. Progressive actions (pass, shoot, drive) are modelled independently over a pitch location value surface and combined in an action likelihood model to produce the overall EPV. The value surfaces included in the model also enable off-ball player contributions to be evaluated. The EPV model provides a more holistic summation and evaluation of match scenarios.

13.5 Dangerousity

Link et al present Dangerousity[29], a model comprising two attacking components that increase value (Zone and Control) and two defensive components that decrease value (Pressure and Density). Each component is modelled separately using geometry, ball/player physics and a scalar field pitch surface to produce output values on a scale between zero and one. Dangerousity may be continuously calculated throughout an entire match to evaluate either team's control of the game, or it may be applied to specific match situations to assign relative value to individual player actions.

[28] http://www.lukebornn.com/papers/fernandez_sloan_2019.pdf

[29] https://journals.plos.org/plosone/article/file?id=10.1371%2Fjournal.pone.0168768&type=printable

14. References & Inspiration

Alexander, D., 2017. Outside the Box: A Statistical Journey through the History of Football.

Bornn, L., Cervone, D., Fernandez, J., 2019. Decomposing the Immeasurable Sport: A deep learning expected possession value framework for soccer. Available online at http://www.lukebornn.com/papers/fernandez_sloan_2019.pdf

Brunton, S., Kutz, N., 2019. Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control.

Cox, M., 2018. The Mixer: The Story of Premier League Tactics, from Route One to False Nines.

Geron, A., 2019. Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow.

Goodfellow, I., Bengio, Y., Courville, A., 2017. Deep Learning.

Guttag, J. V., 2016. Introduction to Computation and Programming Using Python.

Guardian Football Weekly Podcast. https://www.theguardian.com/football/series/footballweekly

Hochreiter, S., Schmidhuber, J., 1997. Long Short-Term Memory. Available online at https://www.bioinf.jku.at/publications/older/2604.pdf

James, G., Witten, D., Hastie, T., Tibshirani, R., 2017. An Introduction to Statistical Learning with Applications in R.

Karpathy, A., 2015. The Unreasonable Effectiveness of Recurrent Neural Networks. Available online at https://karpathy.github.io/2015/05/21/rnn-effectiveness/

Kurt, W. Count Bayesie https://www.countbayesie.com

Lang, S., Link, D., Seidenschwarz, P., 2016. Real Time Quantification of Dangerousity in Football Using Spatiotemporal Tracking Data. Available online at https://journals.plos.org/plosone/article/file?id=10.1371%2Fjournal.pone.0168768&type=printable

Olah, C. Understanding LSTM Networks. Available online at https://colah.github.io/posts/2015-08-Understanding-LSTMs

Rudd, S., 2011. A Framework for Tactical Analysis and Individual Offensive Production Assessment in Soccer Using Markov Chains. Available online at http://nessis.org/nessis11/rudd.pdf

Scott, L., 2020. Packing in the Bundesliga, Total Football Analysis. Available online at https://totalfootballanalysis.com/data-analysis/packing-in-the-bundesliga-data-analysis

Singh, K., Introducing Expected Threat. Available online at https://karun.in/blog/expected-threat.html

Sumpter, D., 2016. Soccermatics: Mathematical Adventures in the Beautiful Game.

Totally Football Show Podcast. https://www.thetotallyfootballshow.com/

Wells, A., 2007. Football & Chess: Tactics Strategy Beauty.