Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической
кибернетики и компьютерных наук

РАЗРАБОТКА ПРОГРАММЫ РАСЧЕТА XG СТАТИСТИКИ ФУТБОЛЬНЫХ КОМАНД

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Аветисяна Севака Юриковича

Научный руководитель
к. ф.-м. н., доцент _____ 05.06.2017 В. Г. Самойлов

Заведующий кафедрой
к. ф.-м. н. _____ 05.06.2017 С. В. Миронов

Саратов 2017

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

API — (от англ. Application Programming Interface) программный интерфейс

Telegram-бот — приложение, написаное для платформы Telegram

xG — (от англ. expected goals) метрика, характеризующая шансы команды в атаке.

ВВЕДЕНИЕ

Программное обеспечение в спорте стали использовать относительно недавно, на рубеже 20-21 века. Первопроходцами были баскетбольные команды, которые использовали программное обеспечение для анализа данных, накопленных по ходу сезона для последующей покупки необходимых игроков в команду.

Модель xG — метод оценки качества шансов, которые команда создаёт в атаке или позволяет создать сопернику у своих ворот. Сама аббревиатура xG так и переводится — «ожидаемые голы». Позднее появилось ещё несколько показателей: xA — «ожидаемые голевые пас», xW/xD/xL — «ожидаемые победы/ничьи/поражени», xPoints — «ожидаемые набранные очки» и xGa/xAa — «допущенные ожидаемые голы/ассисты» — то есть голы и голевые пасы, которые анализируемая команда позволила создать сопернику, xShooting — оценка качества игры нападающих, а так же xStability — стабильность выступления футбольной команды на протяжении определенного отрезка времени.

На основе модели xG строится xG-карта футбольного матча. xG-карта — это схематичное изображение футбольного поля, на которе наносятся все удары команд в конкретном матче с указанием их xG. Каждый удар отображается на карте в виде окружности, размер которой прямо пропорционален xG этого удара. Цвет окружности варируется в завиимости от того, закончился ли удар голом или нет.

Целью данной работы является формальное описание модели xG и вытекающих из нее моделей xPoints, xShooting и xStability, а так же написание бота для мессенджера Telegram, позволяющего генерировать xG-карты футбольных матчей.

Были поставлены следующие задачи:

— Описать модель xG, а так же вытекающие из нее модели xPoints, xShooting, xStability;
— Найти или разработать API, позволяющий получить xG-метрики футбольных матчей;
— Реализовать классы, необоходимые для генерации xG-карт;
— Реализовать telegram-бот для генерации xG-карт футбольных матчей.

1 Описание задачи и ее решения

Так как модель xG лежит в основе xPoints, xShooting и xStability, то в первую очередь опишем именно ее, так как она является фундаментом для всей последующей работы. Идея написания приложения, которое бы имело максимально простой интерфейс и позволяло генерировать xG-карту отдельно взятого футбольного матча, возникла после прочтения статьи Майкла Кейли «Premier League Projections and New Expected Goals» [1]. В первую очередь было необходимо ознакомиться с самой моделью, формулами подсчета xG [2, 3], описанными в статье. На основе этих данных группой так же заинтересованных моделью xG людей был реализован публичный сервис [4], который собирал информацию о футбольных матчах с множества источников в реальном времени, обрабатывал и выдавал конечному пользователю в виде JSON-объекта. Конечно, существуют и другие реализации модели xG [5, 6], но они не учитывают множества факторов. Описание плюсов и минусов конкретных моделей выходит за рамки данной работы.

В ответе от сервиса содержатся общие данные по матчу, такие как итоговый счет, список игроков обеих команд, так и данные конкретных игроков, содержащие позиции на поле, с которых игроком был произведен удар, завершился ли этот удар голом, а так же xG этого удара.

Имея эти данные, можно приступать к процессу отрисовки. За основу была взята карта матча с официальной страницы Майкла Кайли в Twitter [7], представленная на рисунке 1. Промежуточный результат формируется в виде SVG-документа, так как это позволяет описать в векторном формате конечное изображение. На выходе получается изображения высокого качества, но отсюда вытекает минус данного подхода — большой вес файла. Для уменшения времени ожидания отправки изображения telegram-ботом конечному пользователю, было решено сконвертировать векторный документ в растровый. В итоге, получаем изображение с меньшим весом без значительной потери качества. Пример карты представлен на рисунке 2.

Рисунок 1 – Пример карты xG Майкла Кейли

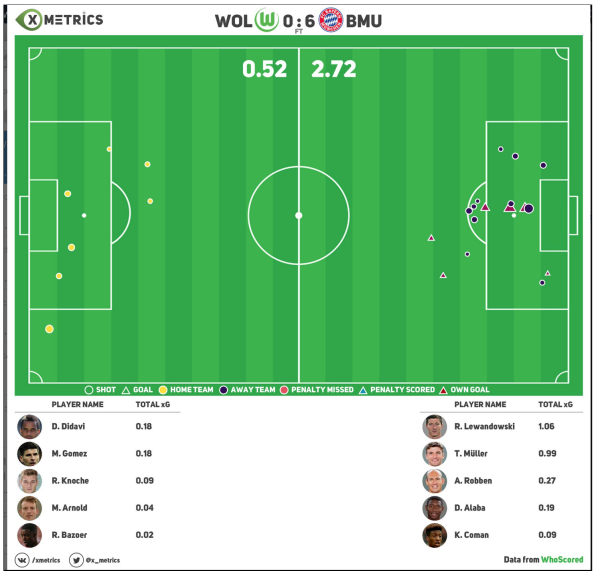

Рисунок 2 – Пример карты xG

2 Описание модели xG

Как было сказано ранее, модель xG — метод оценки качества шансов, которые команда создаёт в атаке или позволяет создать сопернику у своих ворот. В данной главе подробнее рассмотрим то, как считается xG. Допустим, у нас есть случайный эксперимент A — «игрок бьёт по воротам». Какой конкретно игрок, какой частью тела наносит удар, будь то рукой, ногой или головой — пока неважно. У данного эксперимента имеется следующее множество исходов:

— игрок пробил мимо;
— удар заблокирован одним из полевых игроков;
— вратарь парировал удар;
— игрок попал в каркас ворот;
— игрок забил гол.

Есть ещё некоторое количество возможных исходов, таких как «игрок не попал по мячу», но опустим их для простоты понимания. Каждый из этих исходов имеет определенную вероятность наступления, сумма которых даёт нам единицу, то есть 100%-ю вероятность того, что игрок ударил по мячу. Таким образом, вероятность наступления исхода «игрок забил гол» — это и есть значение искомого параметра xG для удара. После матча xG всех ударов суммируется и мы получаем те самые цифры, графики и схемы. Тем не менее нужно помнить, что xG — не совсем классическая вероятность. Её можно интерпретировать как вероятность, но считается как значение функции, сглаженное логистической регрессией [8] (чтобы попадать в диапазон от 0 до 1). То есть, xG — это в первую очередь индекс качества момента, который можно интерпретировать как вероятность забить гол этим ударом. Это очень важный нюанс.

В попытке создать как можно меньше формул, но при этом сохранить нюансы различных видов шансов, автор оригинальной модели xG, Майкл Кейли, остановился на 6 типах ударов по воротам:

— прямые удары со штрафного;
— удары после обыгрыша вратаря;
— удары головой после навеса;
— удары головой после других типов передач;
— удары другими частями тела после навеса;

— так называемые «обычные» удары — удары, нанесённые не головой и после передачи, отличной от навеса.

Автоголы абсолютно случайны, оценить вероятность автогола в матче просто невозможно.

Удары были дифференцированы именно так, потому что множества ударов каждого типа имеют различные (но схожие для ударов одного типа) кривые зависимости вероятности забить гол от угла и расстояния до ворот. Кроме этих факторов, учитываются позиция пасующего и тип атаки.

В первую очередь, проблема отсутствия какой-либо зависимости кроется в объёме выборки — очень важном для статистического анализа понятии. Кейли приводит отличный пример: в предпоследнем сезоне НХЛ [9] Александр Овечкин совершил 795 бросков по воротам, Стефен Карри в том же сезоне 1341 раз бросил мяч в кольцо в НБА [10]. В футболе же есть только два игрока, ударившие по воротам более 1000 раз за последние 6 лет (исключая пенальти).

2.1 Описание модели xShooting

xShooting — это интегрированный показатель, одновременно отражающий следующее:

— насколько хорошо игрок конвертирует опасные моменты в голы;

— насколько качественные моменты он имеет;

— как часто он имеет моменты.

Для начала необходимо определиться с размером выборки N. Необходимо в первую очередь постараться исключить сезонность из выборки; однако это не такая простая задача в силу того, что совокупный объем выборки для анализа очень мал. Итак, для того чтобы не тратиться на подсчет устаревших данных, берутся последние два года выступлений с последнего сыгранного игроком матча для анализа. На этом временном промежутке выбирается не более 150 последних ударов игрока.

Необходимо также понимать, что удары с дальней дистанции несут мало угрозы воротам соперника, поэтому необходимо как-то градировать удары по опасности. Для этого подсчитывается xG каждого удара каждого игрока, количество голов и разбивается на 5 групп, затем берется средневзвешенное конвертации опасности (отношение суммарного количества голов к суммар-

ному количеству xG) по следующим группам со следующими весами:

$$p = \begin{cases} 1/15, & \text{если xG} \leqslant 0.10 \\ 2/15, & \text{если } 0.10 < \text{xG} \leqslant 0.25 \\ 3/15, & \text{если } 0.25 < \text{xG} \leqslant 0.40 \\ 4/15, & \text{если } 0.40 < \text{xG} \leqslant 0.60 \\ 5/15, & \text{если xG} > 0.60 \end{cases}$$

$$\overline{W} = \sum_{i=1}^{5} W_i p_i$$

, где W_i — конверсия на i-ом диапазоне, p_i — вес диапазона.

Результирующее значение затем сглаживается логистической функцией, для того чтобы оно принимало значение в диапазоне от 0 до 1.

Однако это не все. Что отличает действительного топового нападающего от среднего? Все верно — количество ударов. Топовые нападающие имеют больше ударов, чем средние. Пройти мимо такого показателя — значит записать в топ-нападающие любого молодого игрока, который вышел и забил два гола с трех ударов. Но нельзя взять просто количество ударов без ранжирования по опасности. Таким образом, ранжированное количество ударов по воротам вычисляется по следующей формуле:

$$\text{TSR}=0.5\text{n}_1 + 0.75n_2 + 1.15n_3 + 1.6n_4 + 2n_5$$

, где n_i — количество ударов в i-ой группе.

Коэфициент поправки для игрока считается по следующей формуле:

$$A = \begin{cases} \text{TSR} \cdot 0.70/25, & \text{если TSR} < 25 \\ \text{TSR} \cdot 0.85/40, & \text{если } 25 \leqslant \text{TSR} < 40 \\ \text{TSR} \cdot 0.90/55, & \text{если } 40 \leqslant \text{TSR} < 55 \\ \text{TSR} \cdot 0.95/70, & \text{если } 55 \leqslant \text{xG} < 70 \\ \text{TSR} \cdot 1.00/85, & \text{если } 70 \leqslant \text{xG} < 85 \\ \text{TSR} \cdot 1.05/100, & \text{если } 85 \leqslant \text{xG} < 100 \\ \text{TSR} \cdot 1.10/115, & \text{если } 100 \leqslant \text{xG} < 115 \\ 1.15, & \text{если xG} \geqslant 115 \end{cases}$$

Итоговая формула подсчета xShooting имеет вид: $xS = f(\overline{W}) \cdot A \cdot 100$.

Совсем избавиться от сезонности в результатах не получится, однако в дальнейшем планируется аккумулировать эти данные в динамике за последний год, отдавая предпочтение текущему сезону так, чтобы один сильный сезон или выступления в слабой лиге не оказывали существенное влияние на конечный результат. Рассмотрим подробнее данные, полученные для Златана Ибрагимовича, игрока с наибольшим xShooting, равным 94.33. На основе этой таблицы можно сделать определенные выводы и по игре команды в целом.

2.2 Описание модели xPoints

Предположим, что играют команды A и B, команда A создала 10 шансов по 0.2 xG каждый, а команда B — 10 шансов по 0.1 xG. Маловероятно, что матч завершится со счётом 10–10, так же маловероятно, что команда B выиграет 4–0. В теории вероятностей есть формула полной вероятности [11], которая позволяет вычислить вероятность интересующего события через условные вероятности этого события. В случае матча, в котором было создано по 10 моментов с каждой стороны, нужно учесть вероятность наступления каждого потенциально возможного исхода: 0-0, 1-0, 0-1, 1-1 и так далее до 10-10. Затем с помощью формулы полной вероятности необходимо вычислить вероятность победы команды A, вероятность ничьей и вероятность победы команды B. Эти величины нужны, чтобы получить математическое ожидание набранных командой очков, которое считается по формуле $wP \cdot wProb + dP \cdot dProb$,

где:

— wP — количество очков, получаемых за победу;

— $wProb$ — вероятность победы;

— dP — количество очков, получаемых за ничью;

— $dProb$ — вероятность ничьи.

Таким образом мы полуичм примерное количество очков, которое команда должна была набрать за матч.

Тут необходимо понять, что xPoints — это не реальное количество очков, на которое наиграла команда, а некий показатель качества игры. Да, обычные очки, присуждаемые за победу/ничью — это тоже показатель качества, но значение этого показателя распределяется дискретно. Дискретное значение подразумевает собой конечное число вероятностей, тем не менее довольно часто случаются матчи, после которых сложно сказать однозначно, что одна команда была явно сильнее другой. Именно поэтому для моделей, оценивающих качество игры команды, необходимо брать математическое ожидание величины, а не конкретное значение. Оценив разницу реальных очков и xPoints, можно сделать выводы об удаче, сопутствующей команде, и понять, какие команды явно прыгают выше головы, а какие, наоборот, не реализуют свои шансы должным образом.

Для подсчёта xPoints дополним модель xG Кейли и будем учитывать 11-метровые удары. xG таких ударов будет рассчитывается как средняя вероятность конверсии удара в гол по лиге за сезон.

Стоит отметить, что в сумме xW, xD и xL дают ровно количество сыгранных на данный момент матчей, то есть G. Наибольший интерес представляет разница между предпологаемым количеством набранных очков xP и фактическим значением P. Можно заметить, что команда, которая стала чемпионом, имеет довольно существенную разницу $P - xP$, из чего можно сделать вывод, что по ходу сезона команде сопутствовала удача, выигрывалось много матчей, в которых, судя по модели, команда противника была ближе к победе.

2.3 Описание модели xStability

Суть метрики xStability — численное выражение такого понятия, как «стабильность выступления футбольной команды на протяжении определенного отрезка времени».

Команда выступает стабильно, если из матча в матч создаёт у чужих ворот и допускает у своих примерно одно и то же количество шансов. Чтобы придать численное выражение этому параметру, будем отталкиваться от xGD (разницы между созданными и допущенными шансами, $\mathrm{xG} - \mathrm{xGa}$) команды в каждом матче на протяжении исследуемого периода.

В таблицу добавились следующие столбцы:

— Avg. xGD — среднее аримфетическое xGD;

— Max. Δ — максимальная разница xGD двух последовательных матчей;

— Sum. Δ — суммарная Δ;

— Avg. Δ — среднее взвешенное Δ.

Итак, xGD команды в 1 и 2, 2 и 3, 3 и 4 турах и так далее — это и есть дельта, Δ. Логика метода заключается в следующем: если график Δ у команды «скачет» — значит команда от матча к матчу либо создаёт значительно разное количество шансов, либо допускает у своих ворот (либо и то, и другое вместе). Значит, эта команда выступает нестабильно. Здесь нужно уточнить — сама по себе стабильность не может быть положительной и отрицательной характеристикой, все команды можно разделить на 4 вида:

— стабильные и качественно играющие;

— нестабильные и качественно играющие;

— стабильные и некачественно играющие;

— нестабильные и некачественно играющие.

3 Подробное описание решения задачи

3.1 Стек технологий

В данном разделе описаны основные технологии и инструменты, использованные для реализации поставленной задачи.

Основным языком разработки был выбран Java. Java — сильно типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, с помощью виртуальной Java-машины. Данный язык был выбран в связи с огромным количеством написанных на нем прикладных библиотек, а так же наличия опыта работы с ним.

Для разметки xG-карт был выбран векторный формат SVG. SVG (от англ. Scalable Vector Graphics — масштабируемая векторная графика) — язык разметки масштабируемой векторной графики, созданный Консорциумом Всемирной паутины (W3C) и входящий в подмножество расширяемого языка разметки XML, предназначен для описания двумерной векторной и смешанной векторно/растровой графики в формате XML. Векторный формат хранения изображений был выбран с связи с требованием генерации изображений высокого качества, а так же простоты описания и, при необходимости, изменения разметки.

Для конвертации SVG изображения в растровые форматы, такие как PNG, JPEG, использовалась библиотека Apache Batik Transcoder. К сожалению, векторные изображения весят порой слишком много, и пользователи с плохим интернет-соединением вынуждены ждать сгенерированных карт намного дольше, поэтому было решено изображения конвертировать в растровые форматы.

В качестве системы контроля версий использовался Git. Git — распределённая система контроля версиями. Сегодня невозможно представить сколько нибудь серьезный процесс разработки без системы контроля версиями, которые позволяют имет актуальные версии исходного кода в безопасном месте, а так же позволяют синхронизироваться с работой других членов команды. Так же очень удобно иметь возможность «откатиться» на несколько изменений в случае, если какое-то из изменений внесло дефект в работу программы.

Наконец, для хранения итоговых и промежуточных результатов работы была выбрана СУБД MongoDB. MongoDB — документоориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Данная СУБД используется для кеширования промежуточных и итоговых результатов, что ускоряет процесс генерации xG карт. Итоговые xG карты так же складываются в одну из баз MongoDB, что позволяет не генерировать изображения при последующих запросах от других пользователей.

3.2 Разработка клиентского кода для работы с сервисом xG метрик

Для получения необходимой для отрисовки xG карты информации, была написана интеграция с сервисом xMetrics [4]. API данного сервиса реализовано с использованием библиотеки Swagger [12]. Данный подход имеет ряд преимуществ для конечного пользователя, т.к. библиотека генерирует пользовательский интерфейс, позволяющий посмотреть на список доступных методов, а так же попробовать вызвать методы с определенными параметрами, что видно на рисунках 3 и 4.

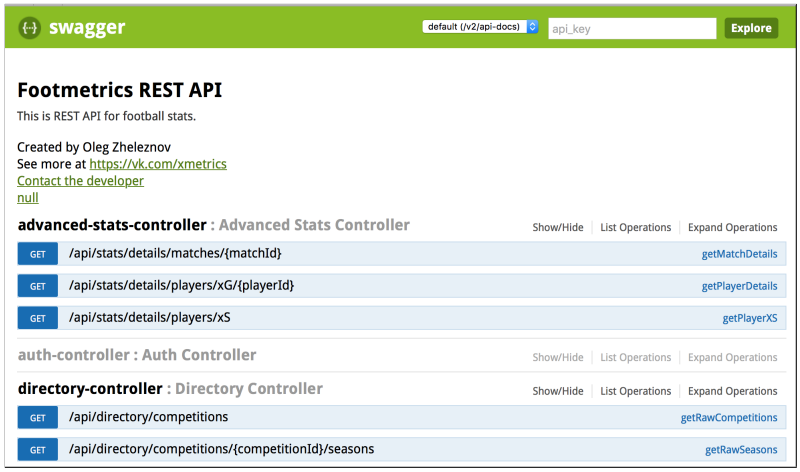

Рисунок 3 – Пользовательский интерфейс Swagger

Были реализованы следующие вызовы к API:

— `/api/directory/competitions` — возвращает список соревнований
— `/api/directory/competitions/{competitionId}/seasons` — возвращает список сезонов конкретного соревнования
— `/api/directory/players/{playerId}/image` — возвращает фотографию футболиста в формате base64

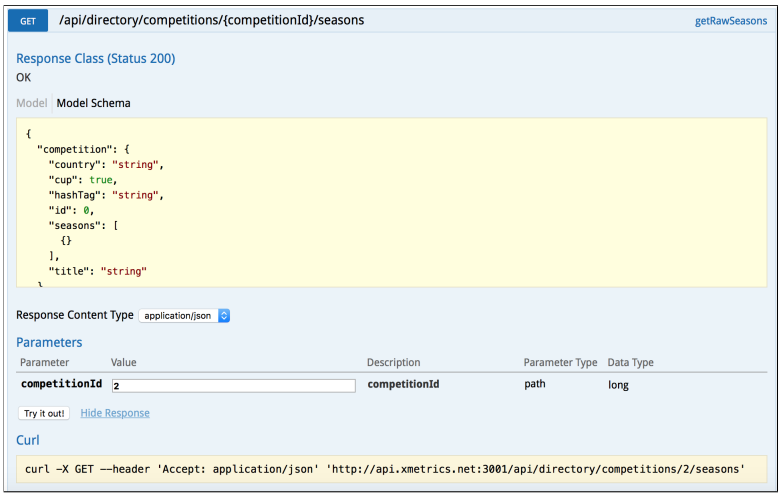

Рисунок 4 – Пример создания запроса в пользовательском интерфейсе Swagger

— `/api/directory/teams/{teamId}/image` — возвращает логотип команды в формате base64
— `/api/directory/seasons/{seasonId}/matches` — возвращает список матчей в выбранном сезоне
— `/api/stats/details/matches/{matchId}` — возвращает полное описание конкретного матча с рассчитанными xG метриками.

3.3 Разработка библиотки для отрисовки xG-карты

Для отрисовки xG карты был написан шаблонизированный svg-документ [13], в который программными средствами вставлялись нужные данные, будь то позиции ударов, или же статистика по каждому игроку в команде. В шаблоне присутствуют переменные вида `${variable}`, где `variable` — один или несколько xml-узлов, описывающих ту или иную часть изображения.

Пример полученной карты представлен на рисунке 2.

Отображаются три типа ударов:

— удар в створ, который не закончился голом
— голевой удар
— пенальти

Неголевой удар на карте отмечен кругом, размер которого прямо пропорционален xG данного удара. Голевые удары отмечены треугольниками. Так же на карте отображен суммарный xG для обеих команд, а так же суммарный xG для каждого из игроков, нанесших удар, если это число больше нуля. На последнем этапе, изображение из векторного конвертируется в растровое.

3.4 Разработка telegram-бота

На этом этапе, имея разработанный клиентский код для взаимодействия с сервисом xMetrics и библиотекой по отрисовке xG карт, остается лишь придумать удобный для пользователя интерфейс взаимодействия с ботом. Подробная информация с примерами кода была взята с официального сайта telegram-бот платформы [14].

Бот имеет следующий список команд:

— `/xg` — генерирует xG карту для конкретного матча, на вход принимает идентификатор матча, который можно узнать с сайта Whoscored [15]

— `/discover` — позволяет пользователю последовательно выбрать соревнование, сезон, матч

Для инициализации бота, обработчики команд необходимо зарегистрировать. Это делается следующим образом:

```
1 bot.registerAll(
2         new XGCommand(drawHandler),
3         new DiscoverCommand(client, drawer, ctx));
```

Рассмотрим подробнее обработчики данных команд в следующем разделе.

3.4.1 Команда `/xg`

Данная команда принимает на вход идентификатор матча. Достаточно много пользователей пользуются сайтом whoscored.com, на котором можно вести обсуждение отдельно взятого матча. Идентификаторы матчей на сайте и в нашей базе данных совпадают, поэтому пользователи сайта могут имея данный идентификатор сгенерировать xG-карту. Данная команды была сделана не в последнюю очередь для удобства отладки бота. Пример работы команды `/xg` можно найти на рисунке 5.

3.4.2 Команда `/discover`

В отличие от команды `/xg`, которая принимает всего лишь один параметр — идетификатор матча, команда `/discover` не принимает параметров, а вся процедура последовательного выбора соревнования, сезона, матча реализована на основе встроенной в платформу telegram-бот клавиатуру [16].

В связи с тем, что пользователей бота может быть много, нужно както различать, от кого боту пришел запрос, а так же на каком этапе команды

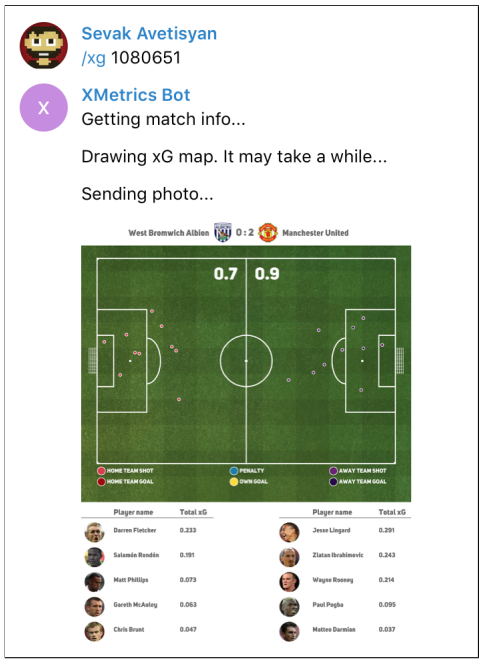

Рисунок 5 – Пример работы команды `/xg`

`/discover` находится данный пользователь. Реализованы методы по изменению и получению данных из контекста, а так же сбросу состояния для выбранных пользователей или чатов. Объект данного класса инициализируется при запуске бота, доступен во всех обработчиках, что позволяет в нужный момент определить дальнейшее действие, которое необходимо выполнить для данного пользователя. Пример работы с командой `/discover` приведен на рисунке 6.

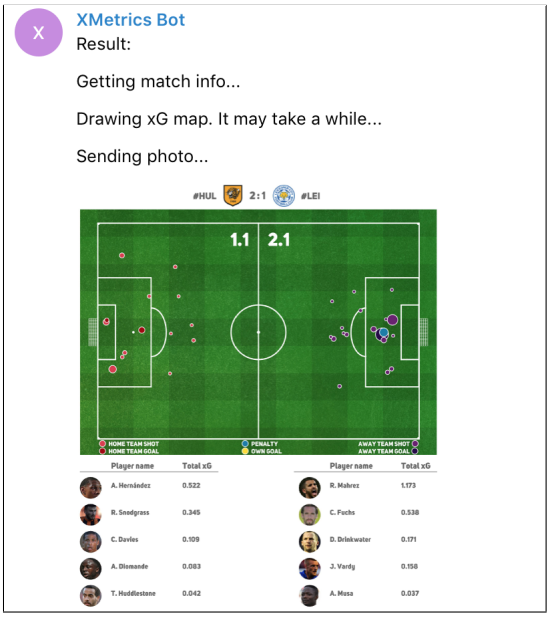

Рисунок 6 – Итоговая xG карта

ЗАКЛЮЧЕНИЕ

В ходе практической работы:

— описаны модели xG, xPoints, xShooting, xStability;
— найден API, ползволяющий получать xG-метрики футбольных матчей;
— реализованы классы, необоходимые для генерации отчета в виде изображений;
— реализован telegram-бот, позволяющий генерировать xG-карты футбольных матчей.

Описанные модели xG, xPoints, xShooting и xStability имеют практическую ценность в первую очередь для футбольных команд, а так же команды аналитиков. Уже сейчас модель xG используется в крупных европейских футбольных клубах, таких как Лондонский Арсенал, Манчестер Сити, Мюнхенская Бавария, а так же в сборной Германии, и с каждым годом ее будет использовать все больше и больше футбольных команд.

Конечно же, xG не может претендовать, да и не претендует, на звание «универсальной теории всего» в футболе. Но если использовать модель с осторожностью и не пытаться судить об игре команды, руководствуясь только xG, она может стать интересным инструментом для оценки качества игры команды, подхода главного тренера к стратегии создания голевых моментов и конверсии шансов — или реализации бьющих игроков.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Premier League Projections and New Expected Goals [Электронный ресурс]. — URL: http://cartilagefreecaptain.sbnation.com/2015/10/19/9295905/premier-league-projections-and-new-expected-goals (Дата обращения 07.05.2017). Загл. с экр. Яз. англ.

2 *Avetisyan, S.* xfootball: what is xg and how to convert xg to xpoints / S. Avetisyan, O. Zheleznov, A. Kuznetcov. — November 2016. http://carrick.ru/blogs/xfootball/.

3 *Zheleznov, O.* xshooting or how to measure attacking players potential / O. Zheleznov. — November 2016. http://carrick.ru/blogs/xshooting/.

4 xMetrics Project [Электронный ресурс]. — URL: http://www.xmetrics.net (Дата обращения 09.05.2017). Загл. с экр. Яз. англ.

5 Deep xG [Электронный ресурс]. — URL: https://deepxg.com (Дата обращения 15.05.2017). Загл. с экр. Яз. англ.

6 xG/xA Dashboards and Data [Электронный ресурс]. — URL: https://differentgame.wordpress.com/xg-shot-maps-and-tables (Дата обращения 15.05.2017). Загл. с экр. Яз. англ.

7 Twitter-аккаунт Майкла Кейли с xG картами [Электронный ресурс]. — URL: https://twitter.com/caley_graphics?lang=en (Дата обращения 07.05.2017). Загл. с экр. Яз. англ.

8 Logistic regression model [Электронный ресурс]. — URL: http://ufldl.stanford.edu/tutorial/supervised/LogisticRegression (Дата обращения 12.05.2017). Загл. с экр. Яз. англ.

9 NHL statistics [Электронный ресурс]. — URL: http://www.nhl.com/stats/leaders (Дата обращения 12.05.2017). Загл. с экр. Яз. англ.

10 NBA statistics [Электронный ресурс]. — URL: http://stats.nba.com (Дата обращения 12.05.2017). Загл. с экр. Яз. англ.

11 Law of total probability [Электронный ресурс]. — URL: https://www.probabilitycourse.com/chapter1/1_4_2_total_probability.php (Дата обращения 12.05.2017). Загл. с экр. Яз. англ.

12 Swagger Rest API Framework [Электронный ресурс]. — URL: http://swagger.io (Дата обращения 12.05.2017). Загл. с экр. Яз. англ.

13 SVG manual [Электронный ресурс]. — URL: https://www.w3schools.com/graphics/svg_intro.asp (Дата обращения 12.05.2017). Загл. с экр. Яз. англ.

14 Telegram Bot platrofm manual [Электронный ресурс]. — URL: https://core.telegram.org/bots (Дата обращения 12.05.2017). Загл. с экр. Яз. англ.

15 Whoscored | Football statistics [Электронный ресурс]. — URL: http://www.whoscored.com (Дата обращения 09.05.2017). Загл. с экр. Яз. англ.

16 Inline Keyboard in telegram bot [Электронный ресурс]. — URL: https://www.botpress.org/docs/telegram/reply-markup-keyboard-inline-keyboard/ (Дата обращения 12.05.2017). Загл. с экр. Яз. англ.

05.06.2017