

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	5
1.1 Основные понятия кластеризации.....	6
1.2 Методы кластеризации данных	15
2 ПРАКТИЧЕСКАЯ ЧАСТЬ	19
2.1 Кластеризация с использованием методов k-means	19
ЗАКЛЮЧЕНИЕ	30
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	30
ПРИЛОЖЕНИЯ.....	33
Приложение А	34

ВВЕДЕНИЕ

Футбол – это один из самых популярных видов спорта в мире. Изучение разных аспектов игры, стилей команд, общих трендов позволяет улучшать этот вид спорта и двигать прогресс вперед.

В современном спортивном анализе возрастает роль количественных методов для объективной оценки выступления команд и игроков. Футбол, как динамичный и сложный вид спорта, предоставляет обширный массив данных, анализ которых позволяет выявлять скрытые закономерности и принимать обоснованные решения.

В данной курсовой работе будут рассмотрены данные по футбольным матчам Английской Премьер-Лиги, потому что она является ведущей лигой мира с высокой конкуренцией команд, инновационными идеями тренеров и наличием высококвалифицированных футболистов. Суммирование всех этих пунктов делает анализ команд актуальным.

Анализ статистических характеристик команд одной из лучшей футбольной лиги может помочь выявить куда направляется вектор развития современного футбола, каким образом удастся достигать результатов, и, наоборот, определить недостатки команд, не позволяющие им побеждать, что является главной целью спорта в общем. Исследование в этой области может помочь получить представление о содержании игр команд.

Цель курсовой работы — Разработка сценария кластеризации методом k-средних на основе данных результатов матчей футбольных команд.

Задачи, решаемые в данной курсовой работе:

- изучение научной и методической литературы;
- определение количества кластеров с помощью метода локтя;
- кластеризация данных с использованием алгоритма k-means;
- использование знаний математической статистики в современной среде обработки данных: аналитической платформы Loginom.

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Основные понятия кластеризации

Изучая стили и игру футбольных команд важно проводить их категоризацию, так как с ее помощью можно увидеть новые ключевые факторы, которые были незначительны при иных методах анализа. То к какой категории относится команда определяет ряд ее статистических характеристик, которые при исследовании показали решающими. Для категоризации было решено провести кластеризацию команд по данным, которые получены в результате сыгранных матчей в регулярном чемпионате.

В данном разделе мы рассмотрим, как осуществляется кластеризация и опишем самые распространённые её алгоритмы.

Сначала определим, что такое кластеризация.

Кластеризация — объединение объектов или наблюдений в непересекающиеся группы, называемые кластерами, на основе близости значений их признаков. В результате в каждом кластере будут находиться объекты, похожие по своим свойствам друг на друга и отличающиеся от объектов, которые содержатся в других кластерах. При этом чем больше подобие объектов внутри кластера и чем сильнее их отличие от объектов в других кластерах, тем лучше кластеризация.

В свою очередь кластер — подмножество объектов статистической совокупности, однородных по своим признакам. В анализе данных кластер — это область, в которой расстояние между объектами меньше, чем до любого объекта вне области (кластера). Например, если рассмотреть людей, то можно выделить типы расы (европеоидная, негроидная, монголоидная, австралоидная).

Элементы кластеризации нередко поступают из небольшого количества логических «источников» или «объяснений», и кластеризация является хорошим способом выявления этих источников.

Кластеризация решает несколько важных задач:

- сегментация данных. Кластеризация позволяет разбить большой набор данных на более мелкие группы, или кластеры, где объекты внутри кластера считаются похожими друг на друга, а объекты из разных кластеров различаются;
- понимание структуры данных. Кластеризация помогает выявить внутреннюю структуру данных, что может быть полезным для анализа и выявления закономерностей;
- сокращение размерности. Кластеризация может использоваться для снижения размерности данных. Вместо работы с множеством признаков, вы можете работать с кластерами, что упрощает анализ;
- рекомендации и фильтрация. В задачах рекомендаций, кластеризация может помочь группировать пользователей или товары на основе их предпочтений, что позволяет предлагать более персонализированные рекомендации;
- обнаружение аномалий. Одним из способов обнаружения аномалий является выявление объектов, не попадающих в какой-либо кластер. Это может помочь выявить необычные или подозрительные данные;
- снижение шума. Кластеризация может помочь выделить схожие группы данных и отфильтровать шум, что улучшает качество анализа.

Кластеризация является одним из наиболее распространенных методов класса методов машинного обучения без учителя. В отличие от методов обучения с учителем, где модель обучается на основе меток классов, методы обучения без учителя не требуют предварительных меток классов для обучения. Вместо этого они стремятся выявить внутренние паттерны, группировать похожие объекты и выполнять другие задачи на основе необученных данных.

Обучение без учителя имеет широкий спектр применений в анализе данных, обработке изображений, обработке текста, биологии, финансах и других областях. Оно позволяет извлекать ценную информацию из необученных данных и находить скрытые закономерности, что делает его мощным инструментом для анализа и понимания данных.

Вообще кластеризация представляет мощные средства для изучения и интерпретации данных, обеспечивая глубокое понимание структур и закономерностей, присутствующих в них.

Она является одним из самых популярных методов анализа и применяется во многих сферах, например:

- маркетинг и реклама. Разбиение аудитории на типы, для создания более направленного и эффективного контента;
- медицина. Классификация пациентов по их анамнезам для диагностики заболеваний, а также анализ данных для выявления паттернов заболеваний;
- финансы. Обнаружение мошенничества в банковских операциях и разбиение портфелей, активов и т.д. по уровню риска;
- аудио и видео. Сегментация звука, видео и изображений.

Есть также множество других областей, в которых применяется кластеризация. Ее часто используют как для типологизации, так и для поиска закономерностей в данных, которые сложно увидеть с первого взгляда.

Обычно перед кластеризацией проводят обработку и очистку данных. Это делается для того, чтобы алгоритмы метода показали более качественный результат.

В обработку данных входит:

- кодирование категориальных признаков;
- нормализация числовых признаков;
- обработка выбросов;
- проверка атрибутов на наличие корреляции.

Кодирование категориальных признаков. Категориальные признаки обычно представлены строковыми значениями, невозможными для прямого использования алгоритмами кластеризации. Но существуют различные методы кодирования, которые позволяют преобразовать категориальные признаки в числовые, чтобы они стали пригодными для анализа.

Например, каждому уникальному значению категориального признака можно поставить в соответствие его порядковый номер. Другой вариант обработки — создание новых признаков, количество которых соответствуют количеству

уникальных значений изначального атрибута, а далее в каждой строке в новых признаках ставится 0 во всех случаях, кроме совпадения значения категориального признака и имени нового (тогда ставится 1).

Нормализация числовых признаков. К числовым признакам обычно применяют нормализацию, чтобы уравнивать их значения и предотвратить искажение результатов кластеризации.

Вот несколько методов нормализации:

1. Десятичное масштабирование (decimal scaling).

В данном методе нормализация производится путём перемещения десятичной точки на число разрядов, соответствующее порядку числа:

$$x'_i = \frac{x_i}{10^n}, \quad (1.1)$$

где n — число разрядов в наибольшем наблюдаемом значении.

Например, пусть имеется набор значений: $-10, 201, 301, -401, 501, 601, 701$. Поскольку $n=3$, то каждое наблюдаемое значение делим на 1000 и получаем: $-0.01, 0.201, 0.301, -0.401, 0.501, 0.601, 0.701$.

2. Минимаксная нормализация.

Несложно увидеть недостаток предыдущего метода: результирующие значения всегда будут занимать не весь диапазон $[0,1]$, а только его часть, в зависимости от наибольшего и наименьшего наблюдаемых значений. Если исходный диапазон мал (скажем, 400-500), то получим, что в результате десятичного масштабирования нормализованные значения будут лежать в диапазоне $[0.4,0.5]$, т.е. его изменчивость окажется очень низкой, что плохо сказывается на качестве построенной модели.

Решить проблему можно путём применения минимаксной нормализации, которая реализуется по формуле:

$$x' = a + \frac{x - x_{min}}{x_{max} - x_{min}}(b - a), \quad (1.2)$$

где a и b — диапазон значений;

x_{max} — максимальное значение;

x_{min} — минимальное значение;

x — элемент, к которому применяется нормализация.

3. Нормализация средним (Z-нормализация).

Недостатком минимаксной нормализации является наличие аномальных значений данных, которые «растягивают» диапазон, что приводит к тому, что нормализованные значения опять же концентрируются в некотором узком диапазоне вблизи нуля. Чтобы избежать этого, следует определять диапазон не с помощью максимальных и минимальных значений, а с помощью «типичных» — среднего и дисперсии:

$$x' = (x_i - \bar{X}) / \delta_x, \quad (1.3)$$

где δ — среднеквадратичное отклонение.

Практически для любого типа анализа нужно проводить обработку выбросов. Выбросы — это значения в данных, которые сильно отличаются от остальных наблюдений. Они могут возникать из-за ошибок измерения, природных аномалий или некорректного сбора данных. Выбросы могут быть как наблюдениями с очень большими значениями, так и с очень маленькими.

Влияние выбросов на алгоритмы кластеризации может быть различным. Например, алгоритмы, основанные на расстоянии между точками, такие как k -средних или иерархическая кластеризация, сильно зависят от расстояния между точками. Единственное выбросное значение может сильно исказить результаты, поскольку оно будет удалено от большинства значений.

Проверка атрибутов на наличие корреляции. Статистическая взаимосвязь двух или нескольких случайных величин (либо величин, которые можно с некоторой

допустимой степенью точности считать таковыми). При этом изменения одной или нескольких из них приводят к систематическому изменению других. Математической мерой корреляции двух случайных величин служит коэффициент корреляции.

Некоторые виды корреляционных связей могут быть положительными или отрицательными (возможна также ситуация отсутствия статистической взаимосвязи, например, для независимых случайных величин). Отрицательная корреляция имеет место, когда увеличение одной переменной связано с уменьшением другой. При положительной корреляции возрастание одной переменной вызывает увеличение другой.

При проведении кластеризации используемые признаки не должны иметь сильной корреляции друг к другу.

Существует несколько коэффициентов корреляции, однако самый простой и популярный — коэффициент корреляции Пирсона.

Критерий корреляции Пирсона позволяет определить, какова теснота (или сила) корреляционной связи между двумя показателями, измеренными в количественной шкале. При помощи дополнительных расчетов можно также определить, насколько статистически значима выявленная связь.

Расчет коэффициента корреляции Пирсона производится по следующей формуле:

$$r_{xy} = \frac{\sum(d_x \times d_y)}{\sqrt{\sum d_x^2 \times d_y^2}}, \quad (1.4)$$

где $d_x = x - \bar{x}$;

$d_y = y - \bar{y}$.

Значения коэффициента корреляции Пирсона интерпретируются исходя из его абсолютных значений. Возможные значения коэффициента корреляции варьируют от 0 до ± 1 . Чем больше абсолютное значение — тем выше теснота связи между двумя величинами.

Для оценки тесноты, или силы, корреляционной связи обычно используют общепринятые критерии, согласно которым абсолютные значения r_{xy} меньше 0.3

свидетельствуют о слабой связи, значения r_{xy} от 0.3 до 0.7 — о связи средней тесноты, значения r_{xy} больше 0.7 — о сильной связи.

Данные этапы обработки данных обязательно должны проводиться перед кластеризацией для того, чтобы улучшить качество результатов.

После предобработки данных нужно выбрать метрику расстояния между объектами. Эта метрика будет определять то, насколько объекты «похожи» и в соответствии со значением будет определяться принадлежность объектов к тому или иному кластеру.

Существует множество метрик, вот лишь основные из них:

1. Евклидово расстояние.

Наиболее распространенная функция расстояния. Она представляет собой геометрическое расстояние в многомерном пространстве. Значение метрики определяется по формуле:

$$\rho(x, x') = \sqrt{\sum_1^n (x_i - x'_i)^2}, \quad (1.5)$$

где x — первый элемент;

x' — второй элемент;

x_i — i -ый атрибут первого элемента;

x'_i — i -ый атрибут второго элемента.

2. Квадрат евклидова расстояния.

Применяется для придания большего веса более отдаленным друг от друга объектам. Это расстояние вычисляется следующим образом:

$$\rho(x, x') = \sum_1^n (x_i - x'_i)^2, \quad (1.6)$$

где x — первый элемент;

x' — второй элемент;

x_i — i -ый атрибут первого элемента;

x'_i — i -ый атрибут второго элемента.

3. Расстояние городских кварталов (манхэттенское расстояние).

Это расстояние является средним разностей по координатам. В большинстве случаев эта мера расстояния приводит к таким же результатам, как и для обычного расстояния Евклида. Однако для этой меры влияние отдельных больших разностей (выбросов) уменьшается (т.к. они не возводятся в квадрат). Формула для расчета манхэттенского расстояния:

$$\rho(x, x') = \sum_1^n |x_i - x'_i|, \quad (1.7)$$

где x — первый элемент;

x' — второй элемент;

x_i — i -ый атрибут первого элемента;

x'_i — i -ый атрибут второго элемента.

4. Расстояние Чебышева.

Это расстояние может оказаться полезным, когда нужно определить два объекта как «различные», если они различаются по какой-либо одной координате. Расстояние Чебышева вычисляется по формуле:

$$\rho(x, x') = \max_i (|x_i - x'_i|), \quad (1.8)$$

где x — первый элемент;

x' — второй элемент;

x_i — i -ый атрибут первого элемента;

x'_i — i -ый атрибут второго элемента.

5. Степенное расстояние.

Применяется в случае, когда необходимо увеличить или уменьшить вес, относящийся к размерности, для которой соответствующие объекты сильно отличаются. Степенное расстояние вычисляется по следующей формуле:

$$\rho(x, x') = \sqrt[r]{\sum_1^n |x_i - x'_i|^p}, \quad (1.9)$$

где x — первый элемент;

x' — второй элемент;

x_i — i -ый атрибут первого элемента;

x'_i — i -ый атрибут второго элемента.

Выбор метрики полностью лежит на исследователе, поскольку результаты кластеризации могут существенно отличаться при использовании разных мер.

В данной курсовой работе использовалась аналитическая платформа Logiном, в которой был использован компонент «Кластеризация». В этом компоненте встроена метрика евклидова расстояния, поэтому в курсовой работе используется именно она.

Подводя итоги, можно сказать, что кластеризация — это популярный метод машинного обучения без учителя, который позволяет выделить в общем объеме объектов группы и распределить каждый в свою.

Для реализации кластеризации необходимо предварительно обработать данные: закодировать категориальные признаки, нормализовать данные, обработать выбросы, проверить атрибуты на корреляцию.

А также перед проведением кластеризации нужно определить метрику расстояния, по которой будет проверяться относится ли объект к тому или иному кластеру.

В анализе данных используются различные методы кластеризации данных. Рассмотрим некоторые из них, изучим как они работают и для каких задач используются.

1.2 Методы кластеризации данных

Существует много различных методов кластеризации данных, но можно выделить две основные классификации алгоритмов:

1. Иерархические алгоритмы.

В результате работы таких алгоритмов мы получаем систему вложенных разбиений, соответствующих различным уровням иерархии. Иерархическое представление удобно при интерпретации результатов, когда требуется информация о различных уровнях кластерной структуры, а также в ситуации, когда точное число искоемых кластеров неизвестно.

Иерархические алгоритмы кластеризации, называемые также алгоритмами таксономии, строят не одно разбиение выборки на непересекающиеся классы, а систему вложенных разбиений. Результат таксономии обычно представляется в виде таксономического дерева — дендрограммы.

Классическим примером такого дерева является иерархическая классификация животных и растений. Среди алгоритмов иерархической кластеризации различаются два основных типа. Дивизимные или нисходящие алгоритмы разбивают выборку на всё более и более мелкие кластеры. Более распространены агломеративные или восходящие алгоритмы, в которых объекты объединяются во всё более и более крупные кластеры.

2. Плоские алгоритмы.

Плоские алгоритмы строят одно разбиение объектов на кластеры.

И второй вариант классификации:

1. Четкие (или непересекающиеся) алгоритмы каждому объекту выборки ставят в соответствие номер кластера, т.е. каждый объект принадлежит только одному кластеру.

2. Нечеткие (или пересекающиеся) алгоритмы каждому объекту ставят в соответствие набор вещественных значений, показывающих степень отношения объекта к кластерам, т.е. каждый объект относится к каждому кластеру с некоторой вероятностью.

Рассмотрим один из самых популярных алгоритмов кластеризации, относящийся к плоским, а именно — метод k-means или к-средних.

Это наиболее простой, но в то же время достаточно неточный метод кластеризации в классической реализации. Он разбивает множество элементов

векторного пространства на заранее известное число кластеров k . Действие алгоритма таково, что он стремится минимизировать среднеквадратичное отклонение на точках каждого кластера. Он хорошо масштабируется для большого количества образцов и используется в широком диапазоне областей применения во многих различных областях.

Алгоритм представляет собой итерационную процедуру, в которой выполняются следующие шаги:

1. Выбирается число кластеров k .
2. Из исходного множества данных случайным образом выбираются k наблюдений, которые будут служить начальными центрами кластеров.
3. Для каждого наблюдения исходного множества определяется ближайший к нему центр кластера (расстояния измеряются в метрике Евклида). При этом записи, «притянутые» определенным центром, образуют начальные кластеры.
4. Вычисляются центроиды — центры тяжести кластеров. Каждый центроид — это вектор, элементы которого представляют собой средние значения соответствующих признаков, вычисленные по всем записям кластера.
5. Центр кластера смещается в его центроид, после чего центроид становится центром нового кластера.
6. Шаги 3 и 4 итеративно повторяются. Очевидно, что на каждой итерации происходит изменение границ кластеров и смещение их центров. В результате минимизируется расстояние между элементами внутри кластеров и увеличиваются междукластерные расстояния.

Остановка алгоритма производится тогда, когда границы кластеров и расположения центроидов не перестанут изменяться от итерации к итерации, т.е. на каждой итерации в каждом кластере будет оставаться один и тот же набор наблюдений.

Преимуществом алгоритма являются скорость и простота реализации. К недостаткам можно отнести неопределенность выбора начальных центров кластеров, а также то, что число кластеров должно быть задано изначально, что может потребовать некоторой априорной информации об исходных данных.

Для решения проблемы неопределенности количества кластеров существует метод локтя, который и был использован в данной курсовой.

Метод локтя — это метод, который мы используем для определения количества центроидов (k) для использования в алгоритме кластеризации k -средних. В этом методе для определения значения k мы непрерывно выполняем итерации от 1 до n (здесь n — гиперпараметр, который мы выбираем в соответствии с нашим требованием).

Шаги реализации метода локтя обычно выглядят следующим образом:

1. Запускается алгоритм кластеризации для разного числа кластеров (обычно от 1 до K).
2. Для каждого числа кластеров вычисляется внутригрупповую сумму квадратов (WCSS), которая представляет собой сумму квадратов расстояний между точками внутри одного кластера.
3. Строится график, где по оси X будет число кластеров, а по оси Y — соответствующая внутригрупповая сумма квадратов.
4. Анализируется график и находится точку, где уменьшение внутригрупповой суммы квадратов замедляется (изгиб локтя). Это значение будет оптимальным числом кластеров для вашего набора данных.

Пример результата работы метода представлен на Рисунке 1.2, где по оси Ox — количество кластеров, а по оси Oy — внутригрупповая сумма квадратов расстояний.

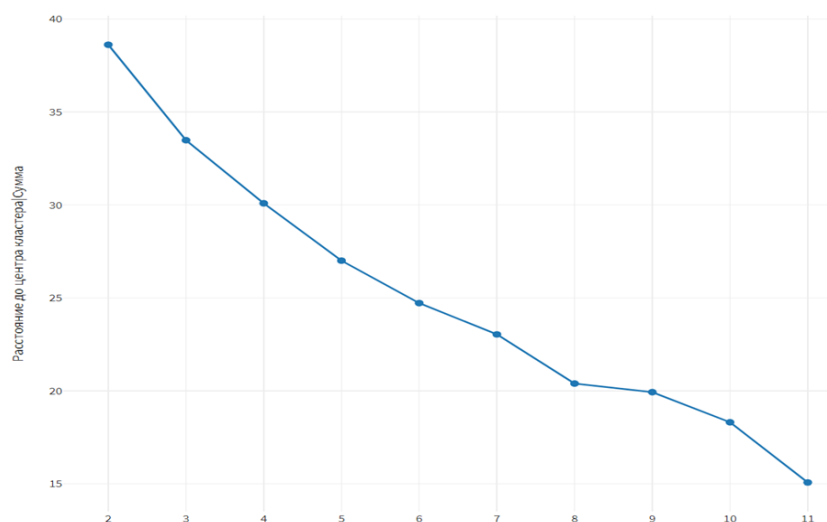


Рисунок 1.2 — График метода локтя

Подводя итоги, можно сказать, что существует множество алгоритмов кластеризации, но наиболее популярный и простой — k-means. Этот метод относится к классу плоских, то есть он строит единственное разбиение объектов на кластеры.

У алгоритма k-means существует определённый недостаток, а именно необходимость знать изначально, сколько кластеров находится в исследуемом наборе данных. Однако для решения данной проблемы существует метод локтя, который является графическим методом оценки количества кластеров. Именно алгоритм k-means и метод локтя используются в данной курсовой работе, что будет продемонстрировано в следующей части.

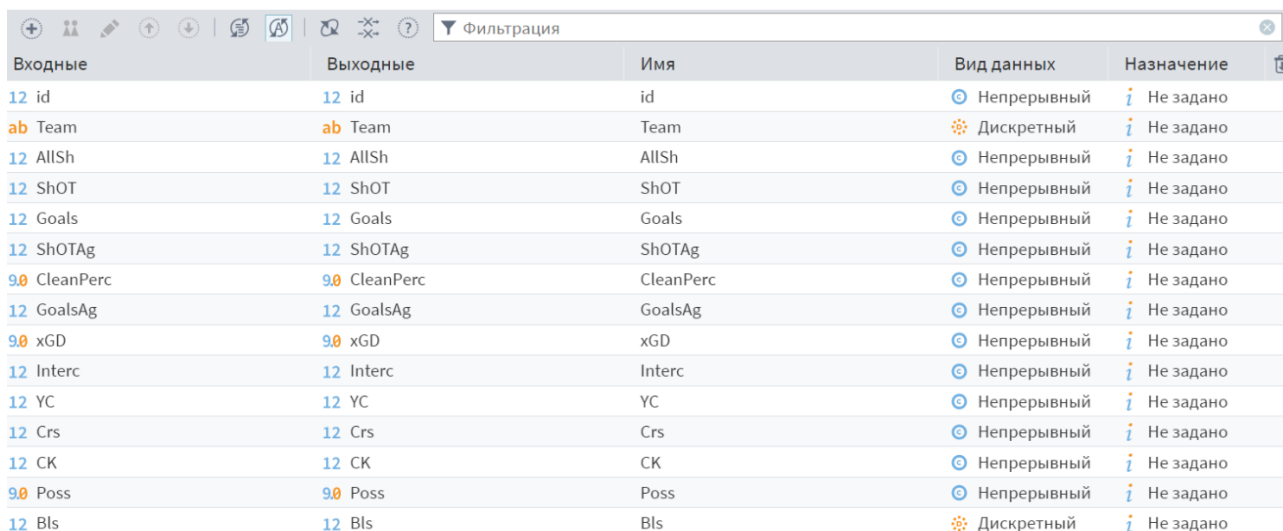
2 ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Кластеризация с использованием методов k-means

При изучении литературы о классификации и кластеризации футбольных команд, а также исследовании исходных данных, выделено 5 характеристик:

- xGD (разность ожидаемых голов от реальных) – очень важный статистический показатель, который определяет качество игры команды в атаке, реализацию опасных моментов;
- количество желтых карточек (YC) – характеристика, показывающая насколько команда дисциплинирована, агрессивность стиля игры;
- количество кроссов (Crs) – характеристика, определяющая стиль игры. Если у команды много кроссов, то она старается играть просто, а это показывает то, что у команды есть сложности в позиционной игре;
- количество перехватов ($Inter$) – этот показатель отражает эффективность игры в защите, а также позволяет оценить эффективность прессинга и работу обороны;
- количество блоков (Bls) – характеристика, указывающая насколько команда сплоченно играет в обороне, однако большое количество блоков может означать проблемы во владении мячом, так как если команда отбивается от атак, то у нее есть проблемы в переходной фазе между обороной и атакой.

Для решения задачи импорта файла добавим в сценарий узел «Импорт текстового файла». В поле имя файлов укажем путь к исходным данным. А также оставим и переименуем те поля, что мы выбрали для анализа. Окно настройки узла импорта показано на Рисунке 2.1.

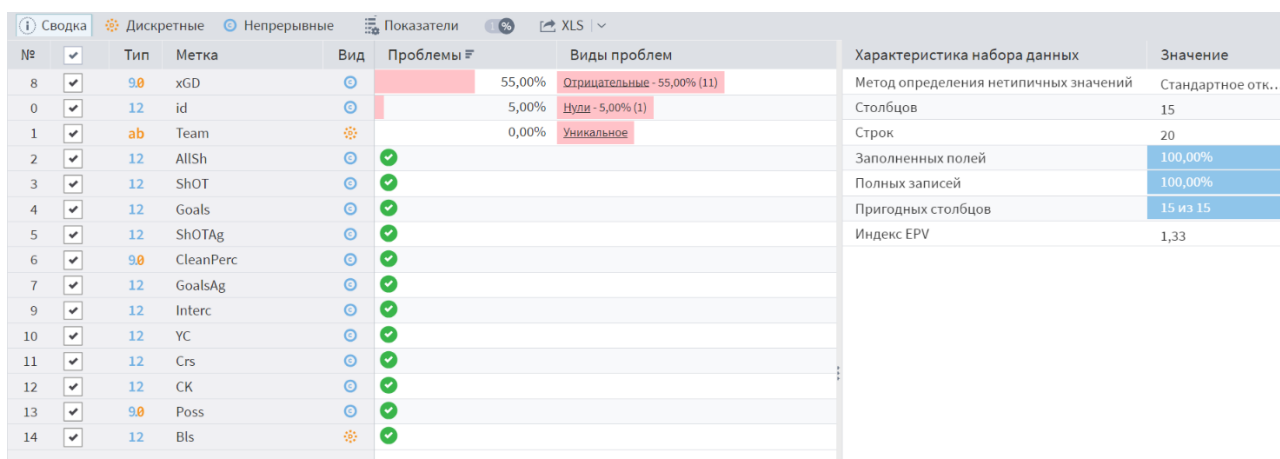


Входящие	Выходные	Имя	Вид данных	Назначение
12 id	12 id	id	Непрерывный	Не задано
ab Team	ab Team	Team	Дискретный	Не задано
12 AllSh	12 AllSh	AllSh	Непрерывный	Не задано
12 ShOT	12 ShOT	ShOT	Непрерывный	Не задано
12 Goals	12 Goals	Goals	Непрерывный	Не задано
12 ShOTAg	12 ShOTAg	ShOTAg	Непрерывный	Не задано
90 CleanPerc	90 CleanPerc	CleanPerc	Непрерывный	Не задано
12 GoalsAg	12 GoalsAg	GoalsAg	Непрерывный	Не задано
90 xGD	90 xGD	xGD	Непрерывный	Не задано
12 Interc	12 Interc	Interc	Непрерывный	Не задано
12 YC	12 YC	YC	Непрерывный	Не задано
12 Crs	12 Crs	Crs	Непрерывный	Не задано
12 CK	12 CK	CK	Непрерывный	Не задано
90 Poss	90 Poss	Poss	Непрерывный	Не задано
12 Bls	12 Bls	Bls	Дискретный	Не задано

Рисунок 2.1 — Импорт данных

Далее необходимо обработать пропуски в столбцах, так как узел «Кластеризация» работает только с полными данными.

Сначала запустим визуализатор «Качество данных» и оценим количество пропусков в полях. Результат работы визуализатора представлен на Рисунке 2.2.



№	Тип	Метка	Вид	Проблемы	Виды проблем	Характеристика набора данных	Значение
8	90	xGD	Непрерывный	55,00%	Отрицательные - 55,00% (11)	Метод определения нетипичных значений	Стандартное откл...
0	12	id	Непрерывный	5,00%	Нули - 5,00% (1)	Столбцов	15
1	ab	Team	Дискретный	0,00%	Уникальное	Строк	20
2	12	AllSh	Непрерывный	✓		Заполненных полей	100,00%
3	12	ShOT	Непрерывный	✓		Полных записей	100,00%
4	12	Goals	Непрерывный	✓		Пригодных столбцов	15 из 15
5	12	ShOTAg	Непрерывный	✓		Индекс EPV	1,33
6	90	CleanPerc	Непрерывный	✓			
7	12	GoalsAg	Непрерывный	✓			
9	12	Interc	Непрерывный	✓			
10	12	YC	Непрерывный	✓			
11	12	Crs	Непрерывный	✓			
12	12	CK	Непрерывный	✓			
13	90	Poss	Непрерывный	✓			
14	12	Bls	Дискретный	✓			

Рисунок 2.2 — Визуализатор «Качество данных»

Как можно заметить, данные не содержат пропусков, следовательно можно продолжить анализ.

Следующим шагом нужно проверить выбранные атрибуты на корреляцию. Так

как при проведении кластеризации исследуемые параметры не должны иметь высокую зависимость друг от друга..

Для этого добавим в сценарий узел «Корреляционный анализ». Поставим галочки напротив используемых характеристик и вычислим для них коэффициент корреляции Пирсона. Для этого зайдём в настройки узла, оставим галочку для флага «коэффициент корреляции Пирсона», так как мы исследуем именно его, а также поставим галочки в столбцах «Набор 1» и «Набор 2» напротив тех полей, которые будут использоваться в кластеризации, а именно — xGD, Interc, YC, Crs, Bls. Нажмём «Далее» и «Выполнить».

На выходе узла получили таблицу с парами полей и коэффициентом Пирсона между ними. Считается, что, если коэффициент Пирсона находится в диапазоне $(-0.5; 0.5)$, то признаки имеют слабую статистическую зависимость друг от друга. Чтобы проверить все ли выбранные признаки подходят добавим в сценарий узел «Фильтр строк» и отберём те строки, в которых значение коэффициента лежит в диапазоне слабой корреляции. Для этого зайдём в настройки узла и нажмём кнопку плюс, чтобы добавить условие. В роли поля будет выступать столбец «Пирсона», в качестве условия поставим знак больше, а значение для сравнения укажем -0.50 .

Нажмём ещё раз на кнопку плюс, чтобы добавить вторую часть условия. В роли поля снова выступает столбец «Пирсона», в качестве условия поставим знак меньше, а значение для сравнения укажем 0.50 . Условный оператор оставим по умолчанию «и». Нажимаем последовательно кнопки «Далее» и «Выполнить». В результате мы получили условие, которое истинно для строк, коэффициент корреляции которых находится в диапазоне от -0.50 до 0.50 .

На выходе узла «Не соответствует условию» получили таблицу, представленную на Рисунке 2.3, в которую попали только пары одноимённых признаков со значением корреляции Пирсона равной единице и 2 пары признаков, имеющие среднюю зависимость, а это значит, что практически все исследуемые поля имеют слабую корреляцию друг с другом, следовательно могут быть использованы в кластеризации.

#	ab Поле1.Имя	ab Поле1.Метка	ab Поле2.Имя	ab Поле2.Метка	9.0 Пирсона ≈ 1
1	Bls	Bls	xGD	xGD	-0,6093519361
2	xGD	xGD	Bls	Bls	-0,6093519361
3	Bls	Bls	Interc	Interc	0,5079028345
4	Interc	Interc	Bls	Bls	0,5079028345
5	xGD	xGD	xGD	xGD	1
6	Interc	Interc	Interc	Interc	1
7	YC	YC	YC	YC	1
8	Crs	Crs	Crs	Crs	1
9	Bls	Bls	Bls	Bls	1

Рисунок 2.3 — Данные, не соответствующие условию

Следующим этапом является кластеризация методом k-means. Сначала необходимо рассчитать количество кластеров. Для этого используем метод локтя.

Реализовывать метод будем в два этапа:

- создание подмодели для одной итерации;
- использование цикла для реализации всех итераций

Для первого этапа создадим узел подмодель. В настройке узла добавим входной и выходной порты для таблицы и для переменных.

Далее настроим переменные. Для этого зайдём в настройки входного порта переменных и добавим два поля с метками «начальное число кластеров K» и «число итераций», именами "k_init" и "iter_count", и установим им значения 2 и 10 соответственно. Тип данных у обоих переменных целый.

Внутри подмодели создадим узел «Кластеризация». Подадим на табличный вход узла таблицу, полученную на входе подмодели. А также покажем порт управляющих переменных и подадим на него переменные полученные на входе подмодели. Выберем все используемые атрибуты и установим им назначение «используемое».

В окне настройки узла уберем флаг «Автоопределение числа кластера» и поставим в число кластеров значение переменной «начальное число кластеров K». Поля раздела «Автоматическое определение числа кластеров» нельзя изменить, так как мы убрали флаг «Автоопределение числа кластеров». Для повторяемости результатов при повторных вычислениях поставим "Random seed" равный 42».

Нажмем сохранить и в настройках режима активации узла (которые находятся

в меню компонента) поставим вариант «всегда переобучать», чтобы нам не приходилось переобучать узел самостоятельно на каждой итерации.

Далее необходимо рассчитать сумму квадратов расстояний до центров кластеров. Добавим в подмодель узел «Калькулятор», подадим на его вход таблицу с верхнего выхода узла «Кластеризация», и рассчитаем в калькуляторе квадрат расстояния для каждой строки. Для этого нажмем значок плюса в окне выражений, тем самым создав новое выражение. Поставим имя "DistanceSquare", а метку «Квадрат расстояния». Тип данных укажем целый, а флаги «промежуточное» и «кэшировать» оставим невыбранными. В теле выражения запишем строку "Distance*Distance", тем самым получив квадрат расстояния. В данном случае поле "Distance" возникло после выполнения узла кластеризация и отвечает оно за евклидово расстояние. Назовем узел «Расчет квадратов расстояния» и выполним его.

С помощью узла «Группировка» вычислим сумму квадратов расстояний, выбрав показателем рассчитанное в калькуляторе поле и методом агрегации «Сумма». Назовем узел «Сумма квадратов расстояний» и выполним его.

Рассчитаем две новых переменных: «Число кластеров» и «Итераций осталось». Для этого добавим узел «Калькулятор (переменные)» в подмодель. Подадим на вход переменные с входного порта подмодели. Зайдем в настройку узла и добавим два новых выражения, нажав два раза на значок плюса. Назовем первое выражение "Clusters" и метку «Число кластеров», а в теле выражения напишем "k_init+1", тем самым обеспечив увеличение количества кластеров при каждой следующей итерации цикла. А второе выражение назовем "Iter2" и метку «Итераций осталось», а в теле выражения напишем "iter_count-1", тем самым уменьшив количество оставшихся итераций. У обоих выражений поставим тип данных целый.

Добавим узел «Переменные в таблицу» в подмодель и подадим на вход входные переменные. В настройке узла выберем вариант «В столбцы».

Добавим узел «Соединение» в подмодель и подадим на вход входные переменные в виде таблицы и сумму квадратов расстояний. В поле «Дополнение до наибольшего набора» выберем вариант «Не дополнять», а в поле «количество строк соответствует» выберем «Максимальному набору».

На выходы подмодели подадим выходы с узла «Соединение» и «Калькулятор (переменные)».

Таким образом, настроена одна итерация кластеризации. Чтобы выполнить ее несколько раз, выйдем из подмодели и добавим в сценарий узел «Цикл» и зайдем в его настройки. В окне выбора узла цикла выберем «Метод локтя итерация», то есть нашу подмодель. В настройке вида узла выберем вид цикла

«Цикл с постусловием», переменную «итераций осталось», условие завершения знак равно, а значение поставим равное нулю. В окне сопоставления переменных свяжем «Начальное число кластеров К» с «Число кластеров» и «Число итераций» с «Итераций осталось». Запустим узел.

Добавим в узел «Цикл» визуализатор «Диаграмма». Результат представлен на Рисунке 2.4, где по оси Ох — количество кластеров, а по оси Оу — сумма квадратов расстояний.

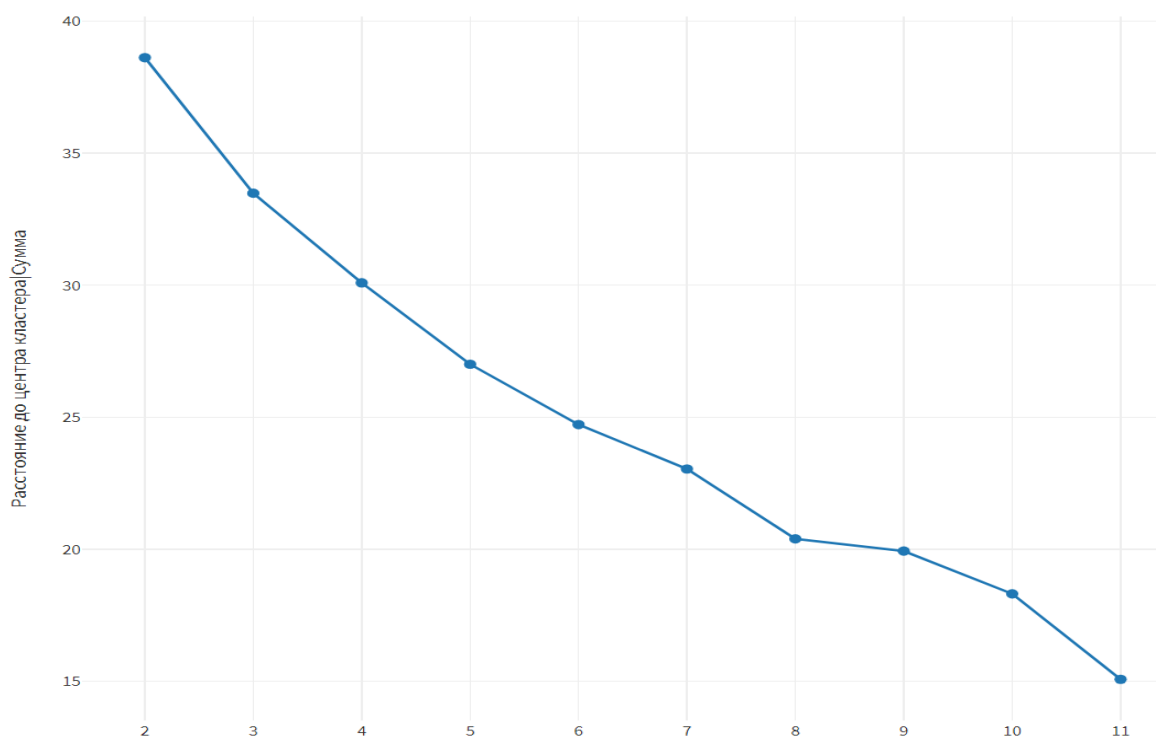


Рисунок 2.4 — Диаграмма метода локтя

Смотря на полученный график, можно заметить, что после 5 кластеров сумма квадратов расстояний уменьшается меньше, чем до 5. Поэтому было принято

решение использовать именно 5 кластеров.

Чтобы реализовать метод k-means, добавим сценарий узел «Кластеризация». Выделим все используемые столбцы и поставим им значение «используемое», в следующем окне уберем галочку напротив «Автоопределение числа кластеров» и поставим число кластеров равное 5.

Далее зайдём в меню и переобучим узел. Нажмём выполнить узел.

Используем визуализатор «Профили кластеров». Результат работы Визуализатор показан на Рисунке 2.5.

#	Метка кластера	Поддержка	xGD	Interc	YC	Crs	Bls
1	Итого	100%	-38,4	233	53	428	353
2	Кластер 4	15%	1,4	236	79	648	389
3	Кластер 2	15%	42,0	233	53	670	353
4	Кластер 0	15%	-38,4	284	101	428	448
5	Кластер 3	20%	-29,7	280	73	550	456
6	Кластер 1	35%	-35,5	325	71	661	437

Рисунок 2.5 — Профили кластеров

Исследуем совместное распределение кластеров по разным атрибутам. Среди наших параметров есть количество перехватов (Inters), количество кроссов (Crs), количество блоков (Bls), количество желтых карточек (YC) и разница ожидаемых голов (xGD).

Полученный результат представлен на Рисунках 2.6–2.10.

На Рисунке 2.6 представлены данные по количеству перехватов.

Атрибут	Итого	Кластер 4	Кластер 2	Кластер 0	Кластер 3	Кластер 1
Значимость	92,0	80	87	83	80	72
Минимум	233	236	233	284	280	325
Максимум	378	288	292	348	331	378
Среднее	315	270	272	310	312	358
Сумма	6308	809	817	930	1247	2505
Стандартное от...	42	29	34	34	22	17
Размах	145	52	59	64	51	53
Пропуски	0	0	0	0	0	0
Значения	20	3	3	3	4	7
Количество ун...						
Центр кластера		270	272	310	312	358

Рисунок 2.6 — Профили по количеству перехватов

На Рисунке 2.7 представлены данные по разнице ожидаемых голов.

Атрибут	Кластер 1	Кластер 3	Кластер 2	Кластер 0	Кластер 4	Итого
🏠 Значимость	63	73	89	74	89	88
📏 Минимум	-35,5	-29,7	42,0	-38,4	1,4	-38,4
📏 Максимум	4,8	-3,3	48,2	16,4	14,6	48,2
📊 Среднее	-9,0	-12,2	45,0	-14,3	6,5	0,0
Σ Сумма	-62,8	-48,9	135,1	-43,0	19,4	-0,2
📏 Стандартное от...	14,3	12,4	3,1	28,0	7,1	24,5
📏 Размах	40,3	26,4	6,2	54,8	13,2	86,6
0 Пропуски	0	0	0	0	0	0
# Значения	7	4	3	3	3	20
S Количество ун...						
△ Центр кластера	-9,0	-12,2	45,0	-14,3	6,5	

Рисунок 2.7 — Профили по разнице ожидаемых голов

На Рисунке 2.8 представлены данные по количеству желтых карточек.

Атрибут	Кластер 3	Итого	Кластер 4	Кластер 2	Кластер 0	Кластер 1
🏠 Значимость	79	92,7	89	80	89	68
📏 Минимум	73	53	79	53	101	71
📏 Максимум	83	109	94	69	109	92
📊 Среднее	78,8	83	87,7	62,0	104	82,7
Σ Сумма	315	1655	263	186	312	579
📏 Стандартное от...	4,6	13	7,8	8,2	4	6,8
📏 Размах	10,0	56	15,0	16,0	8	21,0
0 Пропуски	0	0	0	0	0	0
# Значения	4	20	3	3	3	7
S Количество ун...						
△ Центр кластера	78,8		87,7	62,0	104	82,7

Рисунок 2.8 — Профили по количеству желтых карточек

На Рисунке 2.9 представлены данные по количеству кроссов.

Атрибут	Кластер 3	Кластер 0	Кластер 2	Кластер 1	Кластер 4	Итого
🏠 Значимость	81	83	89	72	89	94,3
📏 Минимум	550	428	670	661	648	428
📏 Максимум	595	578	817	841	671	841
📊 Среднее	580	521	729	731	657	658
Σ Сумма	2322	1564	2187	5115	1970	13158
📏 Стандартное от...	21	81	78	56	13	97
📏 Размах	45	150	147	180	23	413
0 Пропуски	0	0	0	0	0	0
# Значения	4	3	3	7	3	20
S Количество ун...						
△ Центр кластера	580	521	729	731	657	

Рисунок 2.9 — Профили по количеству кроссов

На Рисунке 2.10 представлены данные по количеству блоков.

Атрибут	Кластер 2	Кластер 1	Кластер 4	Кластер 3	Итого	Кластер 0
🏆 Значимость	62	90,4	85	68	91,9	62
📊 Минимум	353	437	389	456	353	448
📊 Максимум	475	504	497	534	534	514
📊 Среднее	413	474	441	489	465	489
Σ Сумма	1240	3319	1323	1955	9305	1468
📊 Стандартное от...	61	25	54	33	44	36
📊 Размах	122	67	108	78	181	66
0 Пропуски	0	0	0	0	0	0
# Значения	3	7	3	4	20	3
S Количество ун...	20	20	20	20	20	20
Δ Центр кластера	353	437	389	456		448

Рисунок 2.10 — Профили по количеству блоков

Внимательно рассмотрев изображения, можно сделать несколько замечаний на счет предназначения кластера и предположить почему узел именно так разбил значения:

- кластер 0. Этот кластер хранит в себе команды, которые показывают грубую игру (большое количество желтых карточек), однако которые пытаются играть в комбинационный футбол, но у них это получается не лучшим образом (маленькое количество кроссов, много пропущенных голов). В этом кластере обитают «упорные, но кризисные» команды.
- кластеры 1 и 3. Эти кластеры показали схожие значения, но в 1 попали команды, которые показали умение хорошо атаковать, однако много пропустили, то есть наблюдается некоторый дисбаланс. В 3 кластере команды пропустили чуть меньше, забили примерно так же, однако стиль игры в данном кластере можно охарактеризовать как позиционный (среднее владение, малое количество кроссов), в 1 же — большое количество кроссов и малое владение, что показывает низкий уровень командной игры. В данных кластерах можем заметить «среднячков», но в 1 кластере, это скорее, команды находящиеся по игре во второй половине турнирной таблицы, а в 3 команды, пытающиеся подняться выше. ;
- кластер 2. В данном кластере состоят команды, которые эффективно атакуют и защищаются (высокие цифры xGD, минимальная разница пропущенных мячей), которые также имеют малое количество желтых карточек, что показывает дисциплинированность в обороне. В этот кластер попали лучшие команды чемпионата («топовые»);

- кластер 4. Здесь характерна неплохая игра в атаке (положительная разница xGD), достаточно дисциплинированные оборонительные показатели, а также сбалансированная игра (владение выше среднего, достаточное количество кроссов). Назовем команды, попавшие в этот кластер таким образом: «интересные, перспективные».

Разработка сценария на аналитической платформе Logiном представлена в Приложении 1.

ЗАКЛЮЧЕНИЕ

В данной курсовой работе был проведен анализ футбольных матчей Английской Премьер-лиги с использованием метода кластеризации k-средних. На основе выбранного набора статистических показателей (разница ожидаемых голов, количество перехватов, количество кроссов, количество желтых карточек и т.д.), были сформированы кластеры команд, отражающие текущее положение дел.

Анализ полученных кластеров позволил выявить закономерности в распределении матчей по их статистическим характеристикам. Были определены ключевые факторы, влияющие на формирование кластеров, и проанализированы их влияние на результат матчей. Например, были выявлены кластеры команд с надежной атакующей игрой (Манчестер Сити (кластер 2), Астон Вилла (кластер 4), с проблемной игрой в обороне (Шеффилд Юнайтед (кластер 3), Лутон Таун (кластер 1), с кризисной ситуацией в команде (Манчестер Юнайтед (кластер 3), Челси (кластер 0).

Результаты исследования показали, что метод k-средних является эффективным инструментом для анализа футбольных матчей и выявления скрытых закономерностей. Однако, важно отметить, что эффективность кластеризации зависит от выбора статистических показателей и количества кластеров. Выбор неудачных параметров может привести к некорректным результатам.

Цель курсовой работы — Разработка сценария кластеризации методом k-средних на основе данных результатов матчей футбольных команд.

Задачи, решаемые в данной курсовой работе:

- изучение научной и методической литературы;
- определение количества кластеров с помощью метода локтя;
- кластеризация данных с использованием алгоритма k-means;
- использование знаний математической статистики в современной среде обработки данных: аналитической платформы Loginom.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1. Сорокин, А. Б. Технологии обучения: кластеризация и классификация: учебное пособие / А. Б. Сорокин, Л. М. Железняк. — Москва: РТУ МИРЭА, 2021. — 49 с. — Текст: электронный// Лань: электронно-библиотечная система. —

URL:<https://e.lanbook.com/book/182493> (дата обращения: 14.12.2024). — Режим доступа: для авториз. пользователей. Кластер (Cluster) / URL: <https://wiki.loginom.ru/articles/cluster.html>

1.2. Стивен С. Скиена. Наука о данных. Учебный курс / Стивен С. Скиена. — Москва: Вильямс, 2020. — 544.

1.3. Джоэл Грас. Data science Наука о данных с нуля. 2-е издание / Джоэл Грас. — Санкт-Петербург: БХВ-Петербург, 2021. — 418.

1.4. Нормализация данных (Data normalization) / URL: <https://wiki.loginom.ru/articles/data-normalization.html>.

1.5. Корреляция (Correlation) / URL: <https://wiki.loginom.ru/articles/correlation.html>.

1.6. Буре, В. М. Теория вероятностей и математическая статистика: учебник / В. М. Буре, Е. М. Парилина. — Санкт-Петербург: Лань, 2022. — 416 с. — ISBN 978-5-8114-1508-3. — Текст: электронный// Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/211250> (дата обращения: 14.12.2024). — Режим доступа: для авториз. пользователей. Обзор алгоритмов кластеризации данных / URL: <https://habr.com/ru/post/101338/>.

1.7. Сорокин, А. Б. Технологии обучения: кластеризация и классификация: учебное пособие / А. Б. Сорокин, Л. М. Железняк. — Москва: РТУ МИРЭА, 2021. — 49 с.— Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/182493> (дата обращения: 14.12.2024). — Режим доступа: для авториз. пользователей. Метод k-средних (K- means) / URL:

<https://wiki.loginom.ru/articles/k-means.html>.

1.8. Elbow Method for optimal value of k in KMeans / URL:
<https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/>.

ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1. Ссылка на набор статистических данных:
<https://fbref.com/en/comps/9/2023-2024/2023-2024-Premier-League-Stats>.

2.2. Компонент «Кластеризация» в Loginom / URL:
<https://help.loginom.ru/userguide/processors/datamining/clustering.html>.

2.3. Визуализатор «Диаграмма» в Loginom / URL:
<https://help.loginom.ru/userguide/visualization/chart/>.

ПРИЛОЖЕНИЯ

Приложение А — графический материал

Приложение А

На Рисунке А.1 представлен основной сценарий проекта в Loginom.

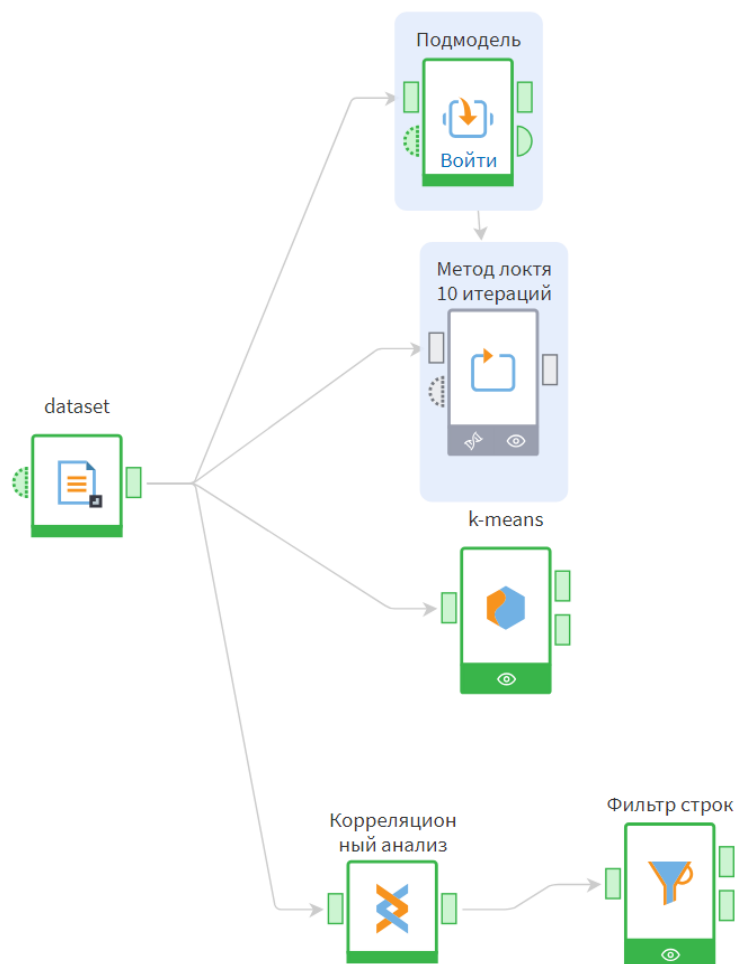


Рисунок А.1 — Основной сценарий Loginom

Описание сценария:

1. Парсинг данных с сайта в файл
2. Импорт датасета.
3. Корреляционный анализ.
4. Фильтр результатов корреляционного анализа.
5. Подмодель одной итерации метода локтя.
6. Цикл 10 итераций метода локтя.
7. Реализация метода k-means.

На Рисунке А.2 представлен сценарий подмодели.

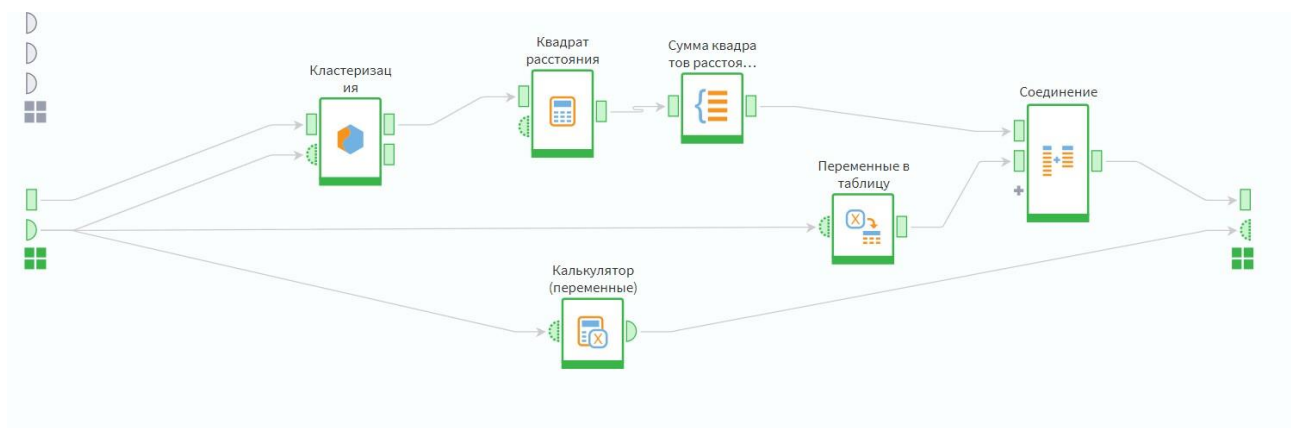


Рисунок А.2 — Сценарий подмодели

Описание сценария:

1. Кластеризация с параметрами, заданными в переменных.
2. Расчет квадрата расстояний.
3. Расчет суммы квадратов расстояний.
4. Расчет новых значений переменных.
5. Трансформация переменных в таблицу.
6. Соединение таблиц переменных и суммы квадратов расстояний.