Таблица 2

Полет с интервалом одно ребро группы БПЛА по маршруту №1 из табл. 2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ЛА №1	1	3	5	7	9	11	1	2	3	4	5	6	7	8	9
ЛА №2		1	3	5	7	9	11	1	2	3	4	5	6	7	8
ЛА №3			1	3	5	7	9	11	1	2	3	4	5	6	7
ЛА №4				1	3	5	7	9	11	1	2	3	4	5	6

Движение БПЛА с интервалом в одно ребро обеспечивает высокую плотность мониторинга, а, следовательно, вероятность обнаружения утечек нефти, коррозий, деформаций труб или посторонних лиц на прилегающей к системе трубопроводов территории значительно повышается.

Список использованных источников

1. Аллилуева Н.В., Руденко Э.М. Методика решения оптимизационных задач по выбору замкнутых маршрутов на графах на основе генетического алгоритма // Известия института инженерной физики. 2017. No2 (44). С. 63-69.
2. Аллилуева Н.В., Дараган А.Д., Ефремов А.А., Руденко Э.М. Математические аспекты применения генетического алгоритма к решению задачи оптимизации на графах // Труды Московского института теплотехники. 2017. Т. 17. Ч. 1. С. 108-117.
3. Краснов М.Л., Киселев А.И., Макаренко Г.И., Шикин Е.В., Заляпин В.И., Эвнин А.Ю. Вся высшая математика. Т.7. М.: КомКнига, 2006, 208 с.

ПРОГНОЗИРОВАНИЕ ИСХОДА СПОРТИВНЫХ МАТЧЕЙ С ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ

Автор: Козлов М., студент 1 курса направления «Информатика и вычислительная техника» филиала «Протвино» ГБОУ ВО МО «Университет «Дубна».

Научный руководитель: к.т.н., доцент Нурматова Е.В., зав. кафедры информационных технологий филиала «Протвино» ГБОУ ВО МО «Университет «Дубна».

Аннотация

В данной работе проводится исследование методов машинного обучения для прогнозирования, а также изучается математический аппарат алгоритмов машинного обучения. На его основе разрабатывается математическая модель, которая прогнозирует результаты футбольных матчей. Также проводится сравнение точности нейронной сети с другими методами прогнозирования(дерево решений, метод наименьших квадратов).

Annotation

This paper investigates machine learning methods for prediction, studying the mathematical apparatus of machine learning algorithms. Based on this, a mathematical model is developed that predicts the results of football matches. The accuracy of the neural network is also compared with other prediction methods (decision tree, least squares method).

Ключевые слова: прогнозирование, исход спортивных матчей, машинное обучение, алгоритмы, дерево решений, метод наименьших квадратов.

Keywords: prediction, outcome of sports matches, machine learning, ML, algorithms, decision tree, least squares method.

Вступление. Машинное обучение в наше время используются для решения многих задач, таких как создание картин, музыки на основе произведений выдающихся художников, музыкантов, синтез человеческой речи, или даже прогноз курса акций на бирже. Всё это возможно благодаря универсальности методов машинного обучения, характерной чертой которых является не прямое решение задачи, а обучение за счёт применения решений множества сходных задач. Для построения таких методов используются средства математической статистики, численных методов, математического анализа, методов оптимизации, теории вероятностей, теории графов и различных техник работы с данными в цифровой форме. Однако, стоит заметить, что несмотря на свою универсальность, точность результатов работы нейросети во многом зависит от выбора исходных данных. Учтём это при разработке.

Одна из сфер, в которой может пригодится машинное обучение – это ставки на спорт, а именно на командные игры (футбол, баскетбол, хоккей, киберспортивные игры). Букмекерские конторы существуют уже очень давно и в последнее годы они на пике своей популярности, что связано прежде всего с доступностью ставок, которая значительно увеличилась с появлением интернета, а также с нынешним положением в мире. Однако стоит понимать, что как букмекеры, так и нейронные сети не могут предсказать результат матча на все 100%, так как в игре играют многие стохастические(случайные) факторы, способные изменить ход игры. Несмотря на это, букмекерские конторы всё же уходят в плюс. Это связано с тем, что аналитики, как правило, используют различные алгоритмы, на основе которых делают выводы об исходе той или иной игры. Поэтому есть смысл в создании нейронной сети, которая автоматизирует поиск подобных алгоритмов.

Подготовка. Как уже говорилось ранее, машинное обучение во многом зависит от исходных данных. Чаще всего эти данные называют **обучающей выборкой.** Она представляет собой совокупность статистических данных, на основе которой машина обучается. В случае обучения с учителем данные поступают и на вход, и на выход.

Ниже показана часть исходной таблицы статистических данных матчей Российской Премьер-Лиги по футболу 2020/2021 [1].

	Часть	Год	Команда	Соперник	Голы	xG	Удары	Удары в створ	PPDA	Победитель	Проигравший
0	1	2020	Химки	ЦСКА	2	2.57	32	7	32.67	ЦСКА	Химки
1	1	2020	Тамбов	Ростов	1	1.07	16	6	24.70	Ростов	Тамбов
2	1	2020	Уфа	Краснодар	3	2.47	29	13	29.10	Краснодар	Уфа
3	1	2020	Арсенал	Ахмат	0	1.26	16	3	17.04	Ничья	Ничья
4	1	2020	Спартак	Сочи	4	3.59	28	8	47.77	Ничья	Ничья

Рисунок 1 - Статистика по матчам РПЛ 2020/2021

Данная таблица состоит из 36 строк и 11 столбцов. Каждая строка - это статистика команды из колонки «Команда» в матче против команды из колонки «Соперник» Каждый столбец указывает на определенный параметр: год, в который был сыгран матч, команды, количество голов и т.д. Особое внимание стоит уделить параметрам xG и PPDA. Футбол – игра, в которой количество забитых голов не пропорционально сыгранному времени, по сравнению с баскетболом. Из-за этого прогнозирование результатов футбольных матчей на основе обычной статистики очень расплывчато. Для этого был придуман специальный критерий xG (от англ. eXpected Goals — ожидаемые голы). В его основе лежит количество совершенных ударов по воротам. Коэффициент xG зависит от расстояния до ворот,

положения мяча относительно ворот, помехам в лице защитников команды соперника и некоторым другим параметрам.

PPDA (от англ. Passes Allowed Per Defensive Action) — футбольный статистический показатель, который позволяет определить интенсивность прессинга в матче. Рассчитывается как количество передач, которое сделала команда с мячом, разделенное на количество действий в обороне соперником. Чем меньше значение PPDA, тем выше интенсивность игры в обороне. Под действиями в обороне подразумеваются отборы, перехваты, неудачные отборы и нарушения правил.

Далее перед обучением необходимо проанализировать имеющуюся таблицу, чтобы отделить полезные данные и избавиться от «мусора». Например, данные о том, в каком году проходил матч, очень слабо влияют на точность прогноза и сильно мешают обучению. Поэтому удаляем такие столбцы как часть и проигравший.

На следующем этапе необходимо закодировать категориальные признаки для того, чтобы машина могла работать с данными в числовом формате. Воспользуемся возможностями библиотеки Scikit-learn.

```
from sklearn.preprocessing import OrdinalEncoder
ord_enc = OrdinalEncoder()
data["Команда"] = ord_enc.fit_transform(data[["Команда"]])
data["Соперник"] = ord_enc.fit_transform(data[["Соперник"]])
data["Победитель"] = ord_enc.fit_transform(data[["Победитель"]])
data["Проигравший"] = ord_enc.fit_transform(data[["Проигравший"]])
data
```

Рисунок 2 - Кодировка категориальных признаков

Также для повышения качества обучения необходимо масштабировать все данные, т.е. привести их к одной метрике. В этом поможет функция StandardScaler [2].

```
from sklearn.preprocessing import StandardScaler
feature_scaler = StandardScaler()
X_train = feature_scaler.fit_transform(X_train)
X_test = feature_scaler.transform(X_test)
```

Рисунок 3 - Масштабирование данных

Последним этапом в подготовке обучающей выборки является разделение данных на тренировочные и тестовые. Оптимально делить в соотношении 7 к 3. Однако в каждом конкретном случае можно варьировать процентное отношение.

Разработка прогнозирующей модели. Прогнозирующая модель – это прототип реально существующего объекта, целью которого является спрогнозировать действия и их результат на основе имеющихся данных об объекте.

В качестве объектов тут взяты футбольные команды. Целью модели является прогноз результатов матчей года между командами Российской Премьер-Лиги.

Основной принцип прогнозирования заключается в суммировании статистических данных команд по каждому параметру таблицы. Затем создам словарь с векторами команд за сезон. Каждая переменная будет отвечать за один статистический параметр. Также дополнительно высчитаю среднее значение владения мячом за матч по следующей формуле:

$averageHandle = \frac{totalHandle}{matches}$, где $totalHandle$ – владение мячом в течение матча.

Далее проведу обучение модели на обучающей выборке. Создам функцию, параметром которого будет номер сезона, то есть год. Объявлю переменные xTrain и yTrain. В них будут храниться матрицы с входными и выходными данными соответственно. Затем загружу статистические данные из таблицы. Для каждого матча модель высчитывает разницу между векторами команд за определенный сезон и записывает в переменную xTrain. В конце

модель присваивает 1 переменной yTrain, если команда выиграет, и 0 - если нет. После обучения модели можно применить выбранные методы прогнозирования.

Дерево решений представляет собой иерархическую древовидную структуру, состоящую из правила вида «Если…, то...». За счет обучающего множества правила генерируются автоматически в процессе обучения.

Построение осуществляется в 4 этапа:

1. выбор атрибута для осуществления разбиения в данном узле;
2. определение критерия остановки обучения;
3. выбор метода отсечения ветвей;
4. оценка точности построенного дерева.

В основе лежит информационная энтропия: $H = -\sum_{i=1}^{n} \frac{N_i}{N} log\left(\frac{N_i}{N}\right)$, где n — число классов в исходном подмножестве, N_i — число примеров -го класса, N — общее число примеров в подмножестве.

Если выбранный атрибут разбиения обеспечивает максимальное снижение энтропии результирующего подмножества относительно родительского, его можно считать наилучшим.

Основная идея метода наименьших квадратов заключается в нахождении коэффициентов линейной зависимости a и b, при которых значение функции от двух переменных x_i, y_i будет наименьшим.

$$F(a,b) = \sum_{i}^{n} \left(y_i - (ax_i + b)\right)^2$$

В качестве коэффициентов выступают входные данные обучающей выборки, а переменных – прогнозируемая величина [3].

Алгоритм логистической регрессии заключается в разделении пространства исходных значений линейной границей на две области, соответствующие классам. Эта граница задается в зависимости от имеющихся исходных данных и обучающего алгоритма.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Результат логистической регрессии всегда находится в интервале [0, 1], что очень удобно для реализации прогнозирования победителя.

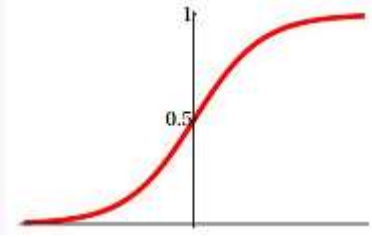

Рисунок 4 - Сигмоида

Нейронная сеть (рис.5) – это последовательность нейронов, соединенных между собой синапсами. Каждый из них связывает два нейрона. Параметром синапса является вес. $H_{\text{вход}} = (l_1 w_1) + (l_2 w_2)$, где H — синапс, l_1 и l_2 – входные нейроны, w_1 и w_2 — веса.

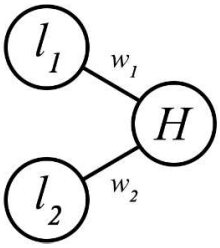

Рисунок 5 - Нейросеть

Для того чтобы получить выходные данные, нужно подставить входное значение в функцию активации. Пропустив число через эту функцию, на выходе получим число в необходимом диапазоне. В качестве функции активации могут выступать линейная функция $(-\infty; +\infty)$, сигмоида (логистическая функция) $[0; 1]$ и гиперболический тангенс $[-1,1]$.

$$H_{\text{выход}} = f_{\text{активации}} * H_{\text{вход}}$$

В качестве проверки достоверности полученных результатов будет использоваться k-кратная перекрестная проверка (cross_val_score). Суть метода заключается в случайном разделении обучающей выборки на k равных частей [4]. Затем каждый такой набор по очереди становится тестовым, а оставшиеся используются для обучения. Результатом проверки является среднее значение всех результатов.

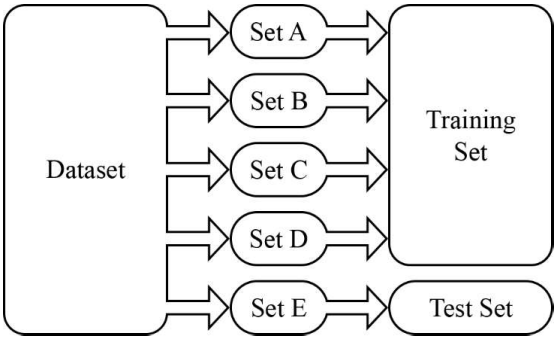

Рисунок 6 - Перекрестная проверка

Реализация модели

1. *Загрузка обучающей выборки.* Подключаем библиотеки для работы с таблицей данных. Загружаем обучающую выборку и список команд.
2. *Удаление команд.* Удаляем команды, по которым мало статистики.
3. *Заполнение векторного словаря.* Затем создаем векторный словарь и заполняем статистикой каждой команды за определенный сезон.
4. *Тренировка модели.* Далее обучаем модель с помощью обучающей выборки.
5. Далее натренированная модель проходит через выбранные методы прогнозирования. Так как все методы давно написаны, используется математическая библиотека Sklearn на основе Python.

Заключение. Представленные данные, полученные вследствие работы нейронной сети, позволяют понять, что машинное обучение как метод прогнозирования спортивных матчей по точности превосходит классические статистические методы.

Список использованных источников

1. Статистические показатели xG для матчей топовых европейских футбольных лиг // understat.com
2. S. Mariam, Machine Learning: Predicting The 2018 EPL Matches / S. Mariam, 2018
3. Т. Рашид, Создаем нейронную сеть — Пер. с англ. / Т. Рашид – СПб.: Диалектика, 2017 – 21 с.
4. Беркинблит М. Б. Нейронные сети. — М.: МИРОС и ВЗМШ РАО, 1993. — 96 с.