ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени М.В.ЛОМОНОСОВА»

ФИЗИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА ОБЩЕЙ ЯДЕРНОЙ ФИЗИКИ

Дипломная работа:

« Оптимизация критериев индентификации частиц в эксклюзивных реакциях электророждения мезонов с использованием алгоритмов машинного обучения »

Выполнил студент

213М группы Голда Андрей Васильевич

Научный руководитель:

к.ф. – м.н., с.н.с. Исупов Е. Л.

МОСКВА

2021

Содержание

Введение

Данная работа выполнена в рамках изучения физики сильного взаимодействия, в которой особый интерес представляют исследования структуры нуклона. Понимание механизмов сильного взаимодействия прольет свет на такие фундаментальные вопросы, как удержание цветных объектов внутри барионов и мезонов, происхождение 98% массы адронов. Детальное описание взаимодействия элементарных конституентов квантовой хромодинамики откроет доступ к механизму формирования сильновзаимодействующих частиц, что особенно важно для понимания начальных стадий эволюции Вселенной.

Нуклонные резонансы – идеальная «лаборатория» для изучения сильного взаимодействия в непертурбативной области, так как многообразие различных возбужденных состояний нуклона позволит наблюдать радиальные, орбитальные и изоспиновые возбуждения с различными квантовыми числами, что позволит в полной мере описать волновую функцию протонов и их возбужденных состояний. Таким образом, изучение нуклонных резонансов является актуальной задачей физики сильных взаимодействий. Изучение нуклонных резонансов доступно из анализа продуктов их распада. В области нуклонных резонансов с массами до *1,5-1,6 ГэВ* доминирующими каналами их распада являются распады на нуклон и пион. В данной работе изучается канал $\boldsymbol{e^- + p \rightarrow e^- + p + \pi^0}$**.** Нейтральный пион, ввиду своего распада за $\sim 10^{-17} c$, может быть реконструирован по двум конечным гамма-квантам. Для регистрации таких эксклюзивных реакций нужен соответствующий детектор, коим в данной работе послужил детектор CLAS12 Национальной Лаборатории им. Т. Джефферсона [3]. Наиболее трудоемкой процедурой является реконструкция нейтральных частиц, в данном случае двух гамма-квантов,

рожденных при распаде нейтрального пиона. Ранее использовались методы, основанные на кинематических отборах, сделанных вручную, которые становятся чрезмерно сложными в многомерном пространстве характеристик конечных частиц. Именно такой мультивариантный анализ может быть эффективно проведен при помощи методов машинного обучения. Данная работа нацелена на создание алгоритма, который мог бы делать такие отборы автоматически с достаточно высокой точностью.

Описание экспериментальной установки

Общая информация

Научная программа детектора CLAS12 очень широка [1] и включает изучение структуры протона и нейтрона как в основном состоянии, так и в их многочисленных возбужденных состояниях, а также в глубоко неупругой кинематике. Спектрометр CLAS12 Национальной Лаборатории им. Т. Джефферсона (Ньюпорт-Ньюс ш. Вирджиния) обеспечивает эффективное восстановление заряженных и нейтральных частиц на большей части полного телесного угла, что в сочетании с непрерывным пучком ускорительного комплекса CEBAF JLAB (*рисунок 1*) создает уникальные условия для изучения структуры нуклона.

Пучок электронов от инжектора ускоряется при помощи уникальной установки [2] с двумя линейными ускорителями, соединенными двумя дугами 180° с радиусом 80 метров. Двадцать криомодулей, каждый из которых содержит восемь сверхпроводящих ниобиевых полостей, образуют два линейных ускорителя. Жидкий гелий поддерживает сверхпроводимость ускоряющих резонаторов при температуре 2° К.

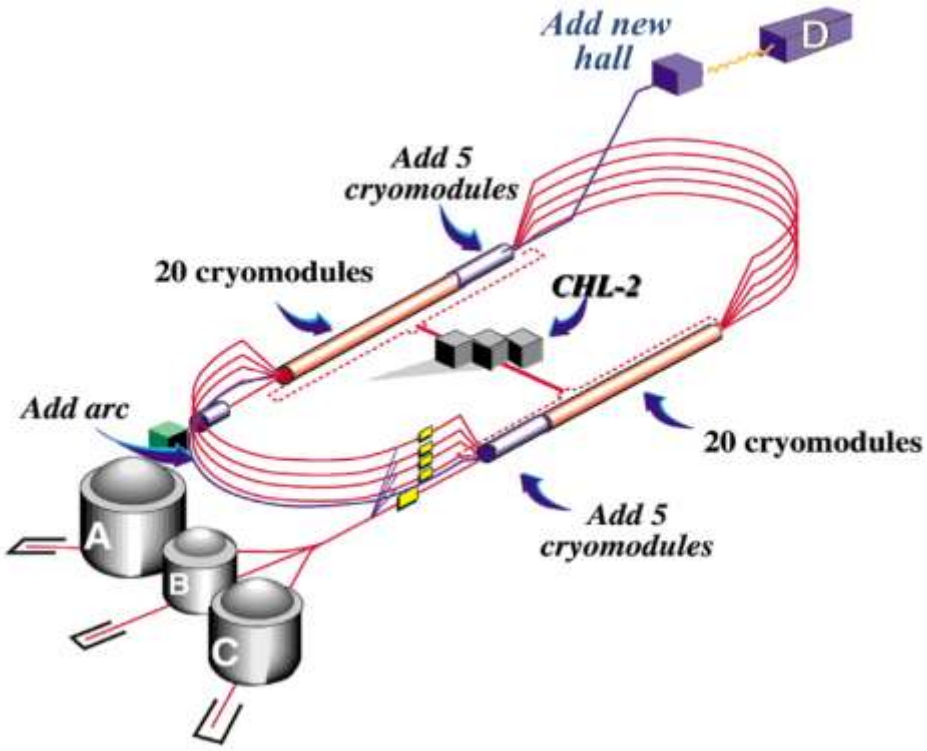

Рисунок 1 Ускоритель непрерывного электронного пучка CEBAF с длиной окружности в 1400 м. Детектор CLAS12 находится в секции B (Hall B)

Данный проект реализовался при финансовой поддержке министерства энергетики США и при сотрудничестве с 48 международными организациями (в том числе НИИЯФ им Д.В Скобельцына), которые внесли свой вклад в проектирование и конструирование аппаратного обеспечения детектора, разработку программного пакета для моделирования сложных схем событий и помощь по введению в эксплуатацию детекторных систем.

Устройство детектора CLAS12

Детектор CLAS12 создавался для изучения процессов, в которых необходимо иметь данные обо всех результирующих частицах в конечном состоянии.

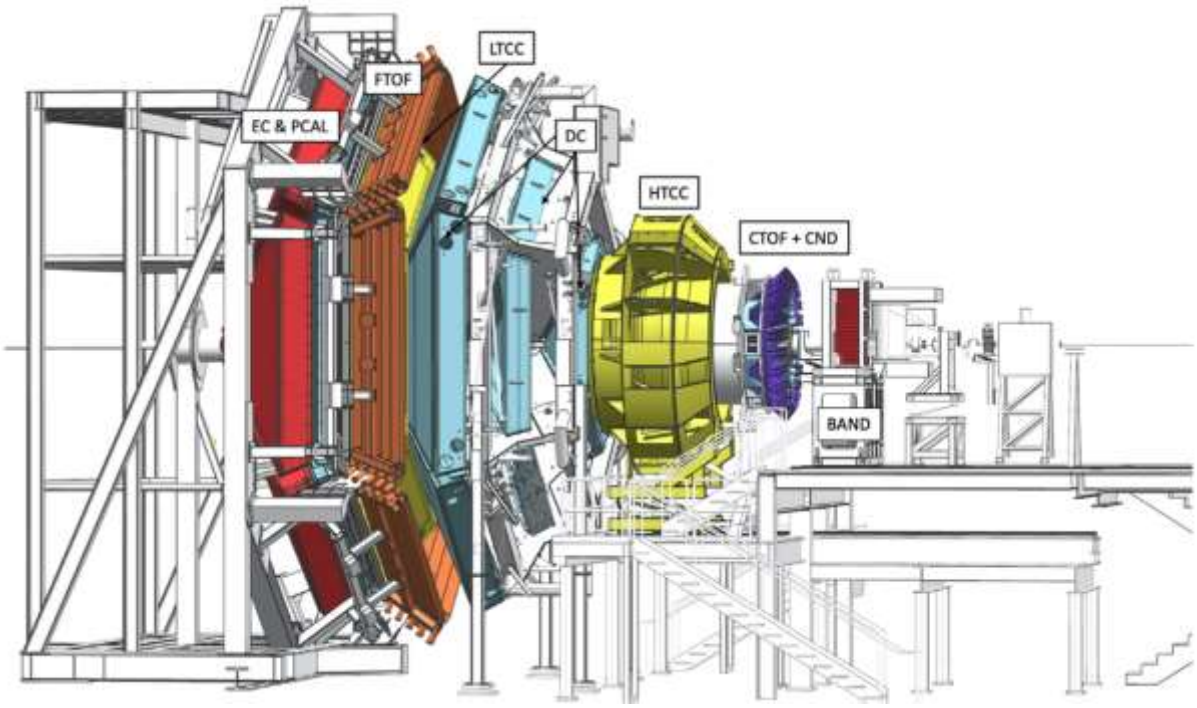

Рисунок 2 Схема детектора CLAS12

Электронный луч входит справа (*рисунок 2*) и взаимодействует с мишенью, в состав которой может входить водород, а также NH_3, ND_3, HD, ${}^{3}_{2}He$ и ${}^{7}_{3}Li$.

Передний детектор

Рассеянные электроны и другие конечные частицы детектируются в переднем детекторе (FD), который состоит из большинства наблюдаемых на схеме детекторов. Высокопороговый черенковский счетчик (High Threshold Cherenkov Counter, HTCC, желтый цвет на схеме *рисунок 2*) с охватом по полярному углу $5\ ^\circ \leq \theta \leq 35\ ^\circ$ и $\Delta\phi = 2\pi$ по азимуту. За HTCC следует тороидальный магнит, система слежения за дрейфовой камерой (DC, голубой цвет на схеме) и еще один набор Черенковских счетчиков (скрытый), времяпролетные сцинтилляционные счетчики (FTOF, коричневый цвет на схеме), электромагнитные калориметры (EC, красный цвет на схеме) [3]. На правом конце установлен нейтронный детектор

обратного угла (BAND, красный цвет на схеме). Весь детектор CLAS12 занимает 13м в пространстве вдоль линии электронного пучка.

Вся конструкция основана на комбинации тороидального магнита с шестью катушками и магнит-соленоида с очень сильным полем (в центральных областях величина индукции магнитного плоя достигает 5 Тл). Комбинированное магнитное поле обеспечивает большой охват как по азимутальному, так и по полярному углам. Реконструкция траектории с использованием дрейфовых камер дает среднее разрешение по импульсу $\frac{\sigma_p}{p} \approx 0.7\%$. При больших полярных углах, когда импульсы частиц обычно ниже 1 ГэВ, среднее разрешение по импульсу составляет $\frac{\sigma_p}{p} \approx 3\%$. Черенковские счетчики, времяпролетные системы и калориметры обеспечивают хорошую идентификацию частиц для электронов, заряженных пионов, каонов и протонов. Быстрый запуск и высокая скорость сбора данных позволяют в течение продолжительных периодов времени работать при светимости $10^{35}\text{с}^{-1}\text{см}^{-2}$. Эти возможности используются в широкой научной программе по изучению структуры и взаимодействий барионов, мезонов и ядер с использованием поляризованных и неполяризованных мишеней.

Рисунок 3 Система дрейфовых камер

Система дрейфовой камеры в системе переднего детектора (FD) CLAS12 состоит из нескольких частей и предназначена для определения координат частицы по времени дрейфа электронов в газе от места ионизации (пролёта частицы) до сигнальных анодных проволок. Малогабаритные камеры R1 расположены прямо перед катушками магнита тора (серый оттенок, *рисунок 3*). Камеры R2 среднего размера зажаты

между катушками магнита, а камеры R3 большого размера расположены сразу после магнита. Шесть катушек тороидального магнита механически поддерживают систему прямого слежения, которая состоит из трех независимых дрейфовых камер в каждом из шести секторов тороидального магнита. Камеры R1 расположены на входе в область магнитного поля тора, камеры R2 расположены внутри магнита, где магнитное поле близко к своему максимуму, а камеры R3 размещены в слабом магнитном поле. Такая компоновка обеспечивает надежную регистрацию заряженных частиц в каждом из шести секторов тора.

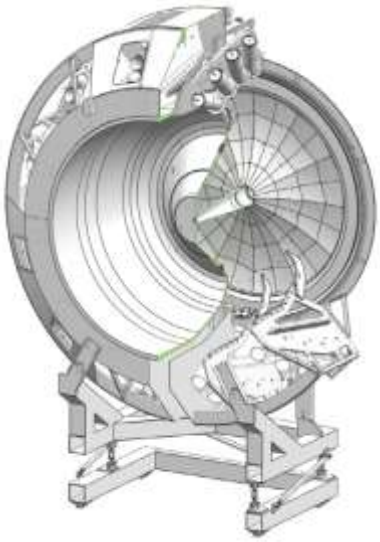

Рисунок 4 HTCC в разрезе

HTCC является основным детектором для отделения электронов (позитронов) с импульсами ниже *4,9 ГэВ* от заряженных пионов, каонов и протонов. Детектор имеет полный охват *360°* по азимуту и диапазон от *5°* до *35°* по полярному углу. Он не имеет слепых зон и, следовательно, покрывает полный телесный угол. Детектор расположен после мишени, зажат между соленоидным магнитом и тороидальным магнитом. Система HTCC должна обеспечивать высокое подавление заряженных пионов и низкий фоновый шум для надежной идентификации рассеянных электронов в плотной электромагнитной фоновой среде. HTCC является отдельной установкой, работающей на сухом газе CO_2 при давлении *1 атм*. Он сконструирован с использованием многофокусного зеркала с 48 эллиптическими зеркальными гранями, которое фокусирует черенковский свет на 48 фотоумножителей (ФЭУ) с кварцевыми окнами диаметром *125 мм*. Детектор имеет диаметр около *4,5 м*. ФЭУ установлены в 12 секторах группами по 4 (*рисунок 4*). Система LTCC (Low threshold Cherenkov counter)

является частью переднего детектора CLAS12 и используется для обнаружения заряженных пионов с импульсами более *3,5 ГэВ*. Система LTCC состоит из ящиков в форме усеченных пирамид и включает в себя 108 легких зеркал и 36 фотоумножителей с конусами Уинстона, которые концентрируют свет, проходящий через относительно большую входную апертуру в меньшую выходную апертуру.

Калориметры (EC и PCAL (pre-shower colorimeter)) в CLAS12 используются в основном для идентификации и кинематической реконструкции электронов, фотонов (например, от распадов $\pi^0 \to \gamma\gamma$ **и** $\eta \to \gamma\gamma$) и нейтронов. PCAL и EC представляют собой калориметры, состоящие из шести модулей [4]. В направлении от мишени EC состоит из двух частей, которые считывают данные отдельно, называемых EC-внутренним и EC-внешним. Они рассчитаны на фиксацию электромагнитных ливней, а также адронных взаимодействий для улучшения идентификации частиц. Каждый модуль имеет треугольную форму с 54 (15/15/24, PCAL / EC-внутренний / EC-внешний) слоями сцинтилляторов толщиной *1 см*, разделенных на полосы шириной *4,5/10 см* (PCAL / EC), зажатых между свинцовыми листами толщиной *2,2 мм*. Слои сцинтиллятора сгруппированы в три группы для обеспечения пространственного разрешения менее *2 см* для энергетических кластеров. Свет от каждой группы считывания сцинтилляторов направляется к ФЭУ по гибкому

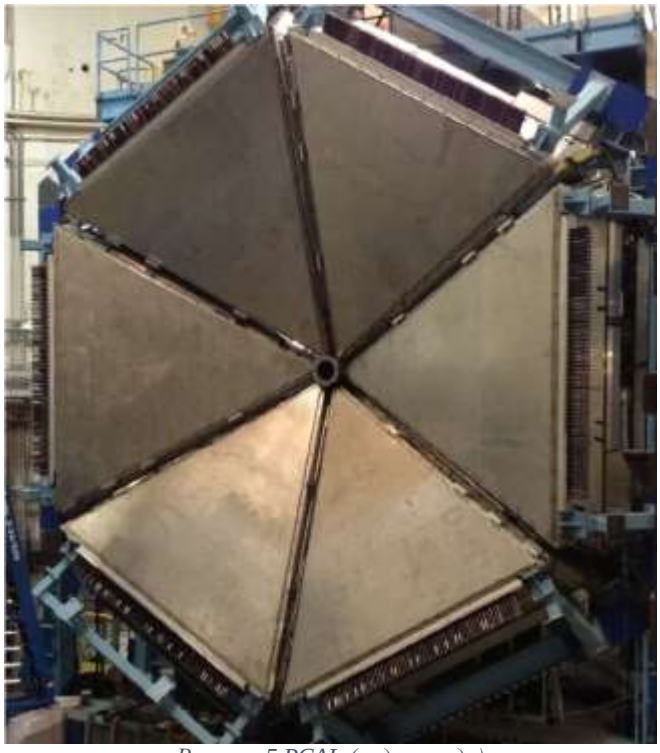

Рисунок 5 PCAL (вид спереди)

оптоволокну. На фотографии (*рисунок 5)* показан калориметр PCAL после установки перед EC.

Частицы, рассеянные от мишени под полярными углами в диапазоне от *35°* до *125°*, обнаруживаются центральным детектором (CD) с его собственными детекторами идентификации и трекинга частиц. Заряженные частицы отслеживаются в CVT (Central Vertex Tracker) и обнаруживаются детектором CTOF (Central Time-of-Flight) с полным охватом *360°* по азимутальному углу. Обнаружение нейтронов обеспечивается центральным нейтронным детектором (CND), расположенным радиально вне CVT и CTOF. Полностью собранный CD показан на фотографии (рисунок 6) после установки в соленоид.

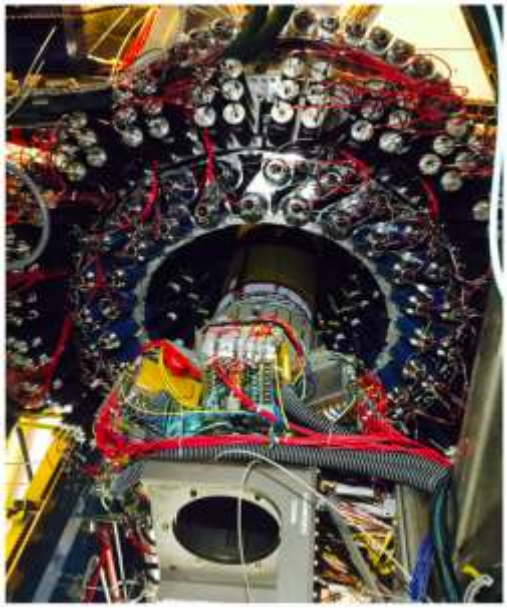

Рисунок 6 Центральный детектор

CVT CLAS12 является частью центрального детектора и используется для измерения импульса заряженных частиц, рассеянных от мишени, которая центрируется внутри соленоидного магнита. Система CTOF используется для идентификации заряженных частиц, вылетающих из мишени, с помощью времяпролетных измерений в диапазоне импульсов от *0,3* до *~1,25 ГэВ*. CTOF (*рисунок 7*) включает 48 пластиковых сцинтилляторов с двусторонним считыванием с помощью ФЭУ через фокусирующие световоды. Набор счетчиков образует герметичный ствол вокруг мишени и CVT.

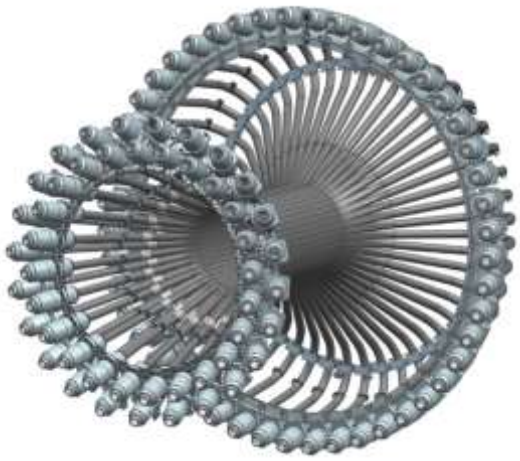

Рисунок 7 Детектор CTOF

Описание методов машинного обучения

Общая информация

Машинное обучение – совокупность методов искусственного интеллекта, которые основаны не на прямом аналитическом решении задачи, а на построении алгоритма за счет решения большого количества схожих коротких задач. Данная область лежит на стыке математической статистики, методов оптимизации и многих классических дисциплин в прикладной и вычислительной математике. Если не брать в рассмотрение дедуктивное обучение, то общий принцип кратко можно описать так: по имеющемуся набору данных восстанавливается неявная зависимость, алгоритм, способный по любому, ранее неизвестному, входному примеру (объекту) дать верный классифицирующий ответ. В данном случае не имеет значения какой класс задач рассматривается: классификация (отнесение объекта к одно из нескольких категорий на основании его признаков), регрессия – (прогнозирование количественных признаков объектов), кластеризация (разбиение множества объектов на группы на основании признаков этих объектов таким образом, чтобы внутри групп объекты были похожи между собой, а вне одной группы – менее похожи) или многие другие. Верный классификационный ответ – это та неизвестная и интересующая нас часть данных, на нахождение которой и нацелено создание алгоритма машинного обучения.

Особенностью машинного обучения является само обучение. Считается, что компьютерная программа учится на опыте $\boldsymbol{E}$ в отношении некоторого класса задач $\boldsymbol{T}$ и показателя производительности $\boldsymbol{P}$, если ее производительность при выполнении задач в $\boldsymbol{T}$, измеренная с помощью $\boldsymbol{P}$, улучшается с опытом $\boldsymbol{E}$ [5]. Набор $\boldsymbol{T}$, $\boldsymbol{P}$ и $\boldsymbol{E}$ может сильно отличаться в

зависимости от задачи, которую мы решаем, но глобально они имеют следующий смысл. Самыми популярными задачами ***T*** в машинном обучении являются уже упомянутые, классификация, регрессия и кластеризация. Опытом ***E*** считаются данные, на основе которых будет построена модель. Данными могут служить изображения, видеоряд, аудиофайлы, текст или таблица с числами. Здесь стоит упомянуть о том, что алгоритмы машинного обучения делятся также на обучение с «учителем» и без него. В задачах обучения без учителя имеется выборка, состоящая из объектов, описываемых набором признаков. В задачах обучения с учителем вдобавок к этому для каждого объекта некоторой выборки, называемой обучающей, известен целевой признак (*рисунок 8*, колонка class) – по сути это то, что хотелось бы прогнозировать для прочих объектов, не из обучающей выборки (*рисунок 9*) [6]. Однако мы сконцентрируемся на рассмотрении задач на основе алгоритмов обучения с учителем, поскольку в данной работе финальную задачу удалось свести к таковой. Метрикой производительности ***P*** алгоритма могут служить различные математические конструкции. Из большого количества метрик (precision, recall, F-мера, AUC-ROC, AUC-PR, logloss, MSE и прочих) мы выделим самую интуитивно понятную метрику – accuracy, которая указывает на долю правильных ответов алгоритма.

Бинарная классификация – задача отнесения элементов набора к двум группам на основе правила классификации. К бинарной классификации можно отнести следующие задачи: по данным о пациенте (давление, уровень гемоглобина и т.п.) определить болен ли пациент; на основе текста письма определить является ли оно спамом; по данным о детали определить является ли она бракованной. В рассматриваемой нами ситуации задача свелась к определению того, произошел данный гамма-квант от распада π^0-мезона, что тоже относится к классу бинарной классификации.

Дадим определение для терминов, которые ниже будут использоваться повсеместно. Для этого мы рассмотрим один из самых популярных примеров в машинном обучении – классификация цветков ириса. Если мы занимаемся обучением с учителем, то у нас неизбежно должна быть обучающая выборка (*рисунок 8*).

	sepal_len	sepal_wid	petal_len	petal_wid
0	5.4	3.7	1.5	0.2
1	4.8	3.4	1.6	0.2
2	4.8	3.0	1.4	0.1
3	4.3	3.0	1.1	0.1
4	5.8	4.0	1.2	0.2
...	...	...	...	...
575	7.7	3.0	6.1	2.3
576	6.3	3.4	5.6	2.4
577	6.4	3.1	5.5	1.8
578	6.0	3.0	4.8	1.8
579	6.9	3.1	5.4	2.1

580 rows × 4 columns

Рисунок 8 Набор из данных к задаче классификации цветков ириса (искусственно сгенерированный)

	sepal_len	sepal_wid	petal_len	petal_wid	class
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

Рисунок 9 Обучающий набор из данных к задаче классификации цветков ириса

В такой выборке каждая колонка будет являться формализованным признаком какого-то объекта, в нашем случае это длина и ширина лепестка и чашелистика. Каждая строчка – это новый объект (пример, instance), в нашем случае это новый цветок. Колонка class (*рисунок 8*) – это целевой признак (target), который нам и необходимо определить на новых примерах (*рисунок 9*), в которых нет этого целевого класса. Здесь важно понять, что обучающий набор содержит целевой признак для обучения модели и создания алгоритма, который в последствии будет использоваться для определения этого целевого признака на данных, в которых он не содержится (изначально не определен и задача требует его определения).

Описание некоторых алгоритмов машинного обучения

KNN

KNN (k-nearest neighbors, метод ближайших соседей) - метод обучения с учителем, в котором мы пытаемся отнести точку данных к определенной категории с помощью обучающего набора. Он является одним из самых простых как в реализации, так и в понимании самого алгоритма.

Как уже было сказано, задача классификации строится на отнесении примера (объекта) к одному из заранее определенных классов на основе его признаков. Поскольку каждый объект представлен в виде вектора в N-мерном пространстве, а также нам известны точные (в обучающей выборке) классы некоторого количества объектов, то возможно просто посмотреть некоторое количество ближайших объектов и по ним сделать вывод о принадлежности к одному из классов, предположив справедливость гипотезы компактности (близкие объекты, как правило, лежат в одном классе). Если формализовать, то алгоритм выглядит следующим образом:

- Посчитать расстояние (считается по-разному, в зависимости от задач) от нового объекта до каждого из объектов обучающей выборки
- Выбрать из этих расстояний ***k*** минимальных (величину ***k*** можно определить “перебором”)
- Классом, в который попадет объект, будет являться тот класс, который чаще всего возникал среди этих ***k*** ближайших соседей

В данном методе существует 2 настраиваемых гиперпараметра (параметра, значения которых задается до начала обучения модели и не изменяется в процессе обучения): метрика расстояния между объектами и количество соседей для определения класса объекта (сюда также можно включить и веса соседей, чем дальше объект, тем с меньшим коэффициентом учитывать его голос). В качестве метрики расстояния могут использоваться: евклидово расстояние, косинусное расстояние, манхэттенское расстояние и многие другие. Чаще всего в качестве начальной метрики используется евклидово расстояние. Число ***k*** в основном определяется через кросс-валидацию (скользящий контроль) в зависимости от того, какое усредненное по всем разбиениям значение (после неоднократного разбиения исходной обучающей выборки на обучающую и контрольную) принимает метрика качества при том или ином значении ***k***. Удобным способом для определения числа ***k*** является построение валидационных кривых (*рисунок 10*). Для задачи с ирисом оптимальными являются ***k*** = 5 (лучше не брать слишком большое или слишком маленькое значение ***k***), если использовать метрику косинусного расстояния [13].

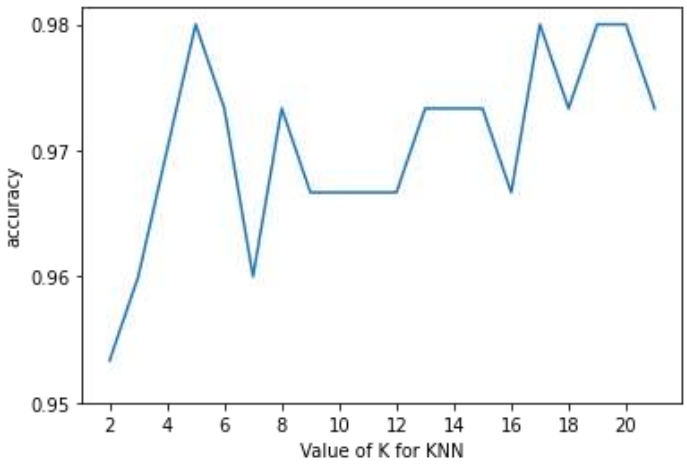

Рисунок 10 Валидационная кривая (зависимость точности модели от значения параметра k)

Данному методу не нужна непосредственная стадия обучения, все сводится к вычислению расстояния между объектами. Исторически KNN считается одним из самых первых и весьма эффективных алгоритмов машинного обучения (например, в канонической задаче с цветками ириса KNN

справляется с точностью выше *97%*). Алгоритм прост как в реализации, так и в интерпретации результатов, однако существуют ряд недостатков, таких как чувствительность к шуму или не относящимся к решению признакам, чувствительность к сильно несбалансированным данным (когда объектов одного класса намного больше, чем объектов другого).

Деревья решений и случайный лес (Decision Trees и Random Forest)

Деревья решений используются в повседневной жизни в самых разных областях человеческой деятельности, порой и очень далеких от машинного обучения. Зачастую дерево решений служит обобщением опыта экспертов, средством передачи знаний будущим сотрудникам или моделью бизнес-процесса компании. Например, до внедрения масштабируемых алгоритмов машинного обучения в банковской сфере и сфере страхования задача скоринга решалась экспертами (*рисунок 11*). В этом случае можно сказать, что решается задача бинарной классификации (целевой класс имеет два значения: "Страховать " и "Не страховать") по признакам "Возраст", "Место эксплуатации", "Езда без аварий", "Стаж" и "Тип автомобиля", где в роли объектов выступают желающие застраховаться. Огромное преимущество деревьев решений в том, что они легко интерпретируемы, понятны человеку, чем выгодно отличается от большинства алгоритмов машинного обучения. Деревья решений получили огромную популярность, а один из представителей этой

Рисунок 11 Пример дерева решений к задаче о выдаче страховки

группы методов классификации, ***C4.5***, рассматривался первым в списке 10 лучших алгоритмов интеллектуального анализа данных в 2008 году [7].

Весь алгоритм строится на подсчете энтропии Шеннона и на последующем пересчете прироста информации [14]:

$$S = -\sum_{i=1}^{N} p_i \log_2 p_i \ ,$$

где $\boldsymbol{p_i}$ – вероятность нахождения системы в $\boldsymbol{i}$-ом состоянии, $\boldsymbol{N}$ – количество возможных состояний системы. Чем ниже энтропия, тем более упорядочена система, что и формализует принцип “наиболее информативного разделения”. Поскольку энтропия – это степень хаотичности системы, то фактическое ее уменьшение – это прирост информации (information gain, IG). Формально, при разбиении выборки по признаку $\boldsymbol{Q}$ прирост информации можно выразить так [14]:

$$IG(Q) = S_0 - \sum_{i=1}^{q} \frac{N_i}{N} S_i \ ,$$

где $\boldsymbol{q}$ - число групп после разбиения, $\boldsymbol{N_i}$ – число элементов выборки, у которой признак $\boldsymbol{Q}$ имеет значение, принадлежащее $\boldsymbol{i}$-ой группе. Построив одно наиболее информативное разделение, можно переходить к следующему, до тех пор, пока дерево не будет построено. Существуют большое количество разных алгоритмов реализации процесса построения дерева, например ***C4.5***, ***CART***, ***ID3***, также существуют вспомогательные методы как для подбора гиперпараметров (критерия разделения: критерий Джини, энтропия; глубины дерева; минимального количества объектов в листовом узле и других), так и для финальной обработки (визуализации, определения точности, уменьшения размеров - pruning). На основе вышеперечисленных критериев можно узнать то, насколько большой вес имеет конкретный признак при построении дерева (найти те признаки, которые наиболее важны). При хорошей точности финальной модели

данный подход сможет уточнить экспертные данные о структуре входных объектов или же просто дать нам новую информацию о физике исследуемой реакции (последнее очень часто получается на практике, когда на вход алгоритму подается объекты из плохо изученной системы, и алгоритм сам показывает наиболее важные признаки).

Случайный лес – алгоритм в машинном обучении, построенный на ансамбле решающих деревьев. Основная идея заключается в том, чтобы использовать большое количество деревьев решений, которые отдельно дают невысокий результат, в целях получения качественной классификации при помощи голосования на их множестве. В основе случайного леса лежат 2 дополнительных алгоритма: первый (бэггинг Бреймана [15]), помогает разбить изначальный тренировочный набор на подпространства в пространстве объектов для того, чтобы использовать их при обучении деревьев решений и валидироваться на out-of-bag подвыборке (подвыборке объектов, которые не попали в обучающую); второй (метод случайных подпространств), который используется для отбора подвыборок со случайными признаками. Оба алгоритма предназначены для уменьшения корреляций между оценками в ансамбле. Процесс построения можно описать в три этапа:

1. Генерация случайных подвыборок с повторением
2. Построение деревьев решений с выбором признаков (построение происходит вплоть до полного исчерпания подвыборки без процедуры прунинга), а само количество деревьев выбирается минимизацией величины ошибки (например, на кросс-валидации)
3. Классификация объектов на основе того, сколько деревьев проголосовало за тот или иной класс (выбирается тот класс, за который проголосовало большинство)

Данный алгоритм (в большинстве случаев) является более точным, чем единичное дерево решений, он также эффективен на данных с большим числом признаков, хорошо поддается распараллеливанию и также имеет возможность оценить важность признаков при классификации.

Бустинг

Модель бустинга была создана для того, чтобы из весьма большого количества относительно слабых моделей (произвольных алгоритмов с точностью лишь немногим отличающейся от случайного угадывания) построить одну сильную. Строится линейная комбинация простых моделей с учетом перевзвешивания входных данных (на каждой последующей итерации с большим весом учитывались те объекты, которые были неверно предсказаны). Иными словами, XGBoost (алгоритма, который использовался в данной работе) основан на идее градиентного бустинга. В основе большинства алгоритмов лежит идея о минимизации функционала для нахождения наиболее точной системы параметров на основе данного набора данных. Целевая функция (функция потерь с регуляризацией) на итерации $\boldsymbol{t}$ (в главном цикле), которую нам нужно минимизировать, следующая:

$$L^t = \sum_i l\left(\hat{y}_i^{(t-1)} + f_t(x_i), y_i\right) + \Omega\left(f_t\right),$$

где l является дифференцируемой функцией потерь, которая измеряет разницу между прогнозом $\hat{y}_i$ и целевым признаком y_i для i-ого объекта; x_i – значения признаков, f_t - функция классификации (в нашем случае – дерево решений, поскольку XGBoost использует именно их в качестве базовых алгоритмов), а $f_t(x_i)$ – предсказание на i-ом элементе. $\Omega\left(f_t\right)$ – функция регуляризации, которая используется с целью предотвращения переобучения (переобучение – явление, когда алгоритм слишком сильно

подстраивается под обучающую выборку и теряет свойство обобщать), ее можно расписать следующим образом: $\Omega(f_t) = \gamma T + \frac{1}{2}\lambda\|w\|^2$, за T можно принять количество вершин в дереве, а за w значения в листьях (γ и λ — параметры регуляризации, которыми как раз можно регулировать "сложность модели"). Такой функционал не может быть оптимизирован с использованием традиционных методов оптимизации в евклидовом пространстве [8], и дальше строятся некоторые аппроксимации на основе разложения Тейлора:

$$L^t = \sum_i l\left(\hat{y}_i^{(t-1)} + g_i f_t(x_i) + 0.5 h_i f_t^2(x_i), y_i\right) + \Omega(f_t),$$

где $g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$, $h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial^2 \hat{y}_i^{(t-1)}}$.

Поскольку мы хотим минимизировать ошибку модели на обучающей выборке, нам нужно найти минимум L^t для каждого t относительно $f_t(x_i)$. Такой минимум (поскольку мы пытаемся постоянно найти минимум для квадратичной функции) будет лежать в $f_t(x_i) = -\frac{g_i}{h_i}$. Каждое дерево ансамбля $f_t(x_i)$ может быть обучено стандартными алгоритмами, которые были описаны выше. Данная модель, как уже видно, включает в себя большое число параметров, которые вшиты в алгоритм и могут быть настроены на начальном этапе инициализации модели, а также, в дальнейшем, подстроены под соответствующие наборы данных.

Данный алгоритм (набор алгоритмов) имеет высокую обобщающую способность, часто используется в бизнес-целях и на соревнованиях (например, на соревнованиях в kaggle.com), не затрачивает большого количества времени на обучение, и, главное, он может быть достаточно легко реализуем, благодаря уже написанным пакетам (например, scikit-learn, написанный на основе языка python).

Применение методов машинного обучения для идентификации гамма-квантов в реакции $e^- + p \rightarrow e^- + p + \pi^0 \rightarrow e^- + p + 2\gamma$ в данных CLAS12

В рамках выполнения данной работы требуется размеченный набор данных, который может быть получен при использовании Монте-Карло моделирования, поскольку в финальной модели мы сами можем отобрать из многочисленных процессов те события, которые соответствуют реакциям, которые мы ищем (реакциям с рождением нейтрального пи-мезона: $e^- + p \rightarrow e^- + p + \pi^0 \rightarrow e^- + p + 2\gamma$). Данный подход возможен, поскольку JLab имеет весь операционный пакет по моделированию процессов, которые протекают в детекторе CLAS12.

Для получения неотобранного набора данных использовался Монте-Карло генератор событий GenKYandOnePion [12], при помощи которого был получен набор событий реакции $e^- + p \rightarrow e^- + p + \pi^0 \rightarrow e^- + p + 2\gamma$ при энергии начального электрона $E_e = 6{,}5$ ГэВ. Сгенерированные события были пропущены через математическую модель детектора GEMC. На этой стадии было 2 набора данных: первый, восстановленные данные с детектора; второй, набор из Монте-Карло моделирования, в котором имелись изначально сгенерированные «правильные» реакции. Далее можно произвести их сравнение, опираясь на то, что если характеристики конечных частиц между наборами схожи (одинаковы с точностью в k процентов), то и восстановленные фотоны с детектора должны происходить от пи-мезонов. Однако остается открытым вопрос о том, по каким характеристикам частиц нужно производить сравнение и как определить эффективную величину зазора (величину k) между данными. Поскольку в нашем распоряжении были характеристики всех частиц в конечном

состоянии, то было принято решение для отбора необходимых гамма-квантов $(\boldsymbol{e^- + p \rightarrow e^- + p + \pi^0 \rightarrow e^- + p + 2\gamma})$, остановиться на их основных характеристиках, а конкретно: $p_{x,y,z}$ - проекциях импульсов гамма-квантов; E_γ – энергии гамма квантов; $\theta_\gamma, \varphi_\gamma, \theta_p, \varphi_p, \theta_e, \varphi_e$ - наборах углов конечных частиц; E_{cal_min}, E_{cal_max}, E_{cal_sum} – энергиях оставленные частицами в калориметре (такие признаки частиц возникают из-за специфики калориметра CLAS12, который состоит из нескольких секторов, где мы выбираем минимальную, максимальную и суммарную энергию по секторам). Коэффициент k подбирался по ряду критериев:

1. k должен иметь физически осмысленное значение, поскольку он отображает степень схожести модельных данных с экспериментальными, и нас не интересуют сильно отличающиеся.
2. Нужно обеспечить достаточный размер финального набора данных (если k будет достаточно маленьким, то размер набора данных сильно уменьшится)
3. Стоит обратить внимание на то, как точность модели (какого-нибудь базового алгоритма) изменится при подборе коэффициента k.

На основе этих критериев подбирался оптимальный коэффициент простой генерацией набора данных при разных значениях k и наблюдались значения размера выборки, точности модели дерева решений (предварительные настройки не проводились, значения параметров: глубины, критериев разбиения и минимального количества значений в листе были выбраны из общих соображений) на полной выборке и на половине получившейся

выборки (этот подход проводился для понимания того, как размер влияет на качество). Результаты приведены на *рисунке 12*.

K	0.01	0.05	0.1	0.15	0.2	0.25	0.50
Dataset size	8489	76258	141294	189900	225526	251810	339522
AS	0.7802	0.8056	0.8283	0.8435	0.8562	0.8635	0.8656
Half set AS	0.7594	0.7494	0.8242	0.8342	0.8524	0.8650	0.865

Рисунок 12 Данные для подбора оптимального значения коэффициента k. Все значения, кроме размера набора данных, указаны в долях.

На основе этих данных было принято решение остановиться на значении $k = 0.2$, поскольку такое значение не искажает физического смысла, размера набора данных достаточно для обучения и точность модели не сильно увеличивается при дальнейшем увеличении параметра, а размер выборки не сильно влияет на точность базовой модели.

В JLab используется специфичные системы хранения данных в формате hipo-файлов, и для конвертации набора данных в привычные для машинного обучения и анализа данных .xlsx, .csv, .json нужно писать определенные программные блоки на языки C++ (в которых данные итеративно достаются из определенных структур файла с данными, фильтруются, сравниваются с модельными). Данный программный блок был успешно реализован, и, в итоге, после сравнения данных с детектора и модельных данных мы можем получить размеченный набор (*рисунок 13*). Программные файлы можно найти по ссылке [9].

	px	py	pz	E	Theta	Phi	E_e	e_theta	e_phi	E_p	p_theta	p_phi	E_cal_max	E_cal_min	E_cal_sum	Cal_n	target_class
0	-0.011190	-0.069473	0.432865	0.438637	0.161122	-1.730490	1.34267	0.153375	-1.142720	1.32363	0.630086	1.197360	0.281400	0.017523	0.432486	4	0
1	-0.007536	-0.028752	0.070961	0.076825	0.396718	-1.827140	1.34267	0.153375	-1.142720	1.32363	0.630086	1.197360	0.281400	0.017523	0.432486	4	0
2	-0.091982	-0.008109	0.241009	0.258093	0.365881	-3.053680	5.50724	0.194536	-0.881099	1.52628	0.828892	2.064330	0.712507	0.052053	1.365600	3	0
3	-0.024211	-0.023357	0.156306	0.159886	0.211991	-2.374160	5.68061	0.167850	-2.314480	1.61430	0.865241	0.846270	0.804000	0.055707	1.415770	3	0
4	-0.016052	-0.017505	0.126329	0.128542	0.165840	-2.312930	5.68061	0.167850	-2.314490	1.61430	0.865241	0.846270	0.804000	0.055707	1.415770	3	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
225521	-0.032847	0.058530	0.295406	0.302935	0.223411	2.082200	5.11566	0.197431	-1.916020	1.52656	0.934634	1.209140	0.700156	0.016319	1.348530	3	1
225522	0.126953	0.013196	0.531980	0.547078	0.235478	0.103589	5.41603	0.148235	-1.283180	1.30956	1.055870	1.976770	0.615615	0.069589	1.222650	3	1
225523	-0.035595	0.035353	0.276175	0.280095	0.179694	2.359610	5.41603	0.148235	-1.283180	1.30956	1.055870	1.976770	0.615615	0.069589	1.222650	3	1
225524	-0.006430	-0.050034	0.084321	0.098259	0.539139	-1.698610	5.74440	0.134487	2.109410	1.38096	0.786543	-0.589201	0.953783	0.056279	1.488490	3	1
225525	0.048318	-0.041591	0.261097	0.268236	0.223027	-0.710718	5.73249	0.120405	0.264206	1.18003	1.121980	-3.014370	0.885641	0.058108	1.511410	4	1

225526 rows × 17 columns

Рисунок 13 Набор данных для обучения, последняя колонка target_class соответствует целевому признаку.

Поскольку главной задачей данной работы является построение алгоритма бинарной классификации, то необходимо определиться с выбором метрики качества построенных моделей. Метрикой был выбран процент верно предсказанных классов для объектов (метрика accuracy). И для того, чтобы в дальнейшем оставалась возможность опираться на нее, не отвлекаясь на несбалансированность, можно искусственно сбалансировать выборку по классам. Данный подход не повлияет на качество модели, поскольку мы увидели, что снижение размера (даже в 2 раза) слабо сказывается на точности (данные можно свободно отсекать, поскольку это просто различные реакции и корреляций между ними нет). На *рисунке 13* представлен финальный набор данных, в котором одинаковое количество объектов с классом 0 и 1 (где классу 1 соответствуют гамма-кванты, которые произошли при распаде пи-мезона, а классу 0 те, которые произошли от других реакций).

После того, как мы подготовили набор данных и определились с метрикой качества, можно переходить к выбору моделей. Начальный подход подразумевал выбор простой модели, ее предварительную настройку, оценку качества и переход к более сложной поочередно (слева направо на *рисунке 14*). Первые простые модели (классификатор на методе ближайших соседей) дали невысокие метрики (0.85), однако показали, что классификация возможна с финальным адекватным результатом, что

позволило в дальнейшем перейти к построению более сложных и точных моделей.

Рисунок 14 Некоторые модели, которые были реализованы в работе, отсортированные по увеличению точности/сложности.

Дальнейшим пунктом исследований является большой блок обучения моделей и дальнейшего выстраивания алгоритмов (в рамках данной работы не строились нейронные сети, поскольку процесс их обучения является трудоемким, а ожидаемого качества мы можем добиться и на классических алгоритмах). Поскольку было построено большое количество моделей и подход к настройке у каждой модели разный, то здесь приводится лишь конечные результаты и краткое описание процесса обучения модели, которая будет использоваться в дальнейшем как основная (ознакомиться с другими моделями можно здесь: [9]).

После обучения моделей в данной работе получились следующие результаты (*рисунок 15*). Проверка на кросс-валидации отличается от проверки на тестовом (отложенном) наборе тем, что при кросс-валидации вся выборка разбивается несколько раз на обучающую и тестовую, и финальным результатом является усредненное значение, а при оценивании на отложенной выборке один раз проводится разбиение, и мы смотрим результаты на одной тестовой выборке. Здесь видно, что наибольшей точностью (accuracy) обладает бустинговая модель (модель

	CV	TTSplit
KNN	0.8512	0.8507
DTree	0.8756	0.8656
RF	0.8712	0.8660
XGBoost	0.9283	0.9149

Рисунок 15 Финальные метрики некоторых моделей после обучения. На кросс-валидации(CV) и тестовом наборе(TTSplit)

XGBoost), что объясняется тем, что ансамблевые модели чаще дают большую точность. Однако при дальнейшем рассмотрении, мы остановились на модели деревьев решений. Хоть ее точность и уступает более сложным моделям, но здесь нам более важна адекватность результатов модели и ее интерпретируемость (поскольку финальный алгоритм может дать нам понимание сути процесса, нас в меньшей степени интересует черный ящик, который возвращает метку класса без пояснения такого выбора). Деревья решений относятся к классу легко интерпретируемых алгоритмов.

Весь процесс построения проводился при помощи библиотеки scikit-learn, написанной на языке python (и предназначенной для использования на языке python). Данная библиотека сильно упрощает процесс обучения, поскольку не нужно тратить время на разработку алгоритмов, нужно лишь правильно подобрать параметры и в дальнейшем получить готовую модель. Основными параметрами, которые было нужно подобрать для построения дерева решений, были (в модели подбирались не все возможные, полный лист параметров можно найти [11]):

- Критерий разбиения - функция измерения качества разбиения. Перебираемые значения: критерий Джини и Энтропия
- Глубина - максимальная глубина, которую дерево может достичь при построении. Перебираемые значения: от 1 до 40.
- Минимальное количество объектов, необходимое для разделения внутреннего узла. Перебираемые значения: от 1 до 5.
- Количество признаков, которые следует учитывать при поиске лучшего разделения. Перебираемые значения: 2,4,6,8,17.

Список параметров, как и перебираемые значения подбирались из общих соображений на основе значений, применяемых в других, не связанных с

данной задачей, моделях. Параметры, используемого для применения этого метода, оптимизируются путем “жадного” перекрестного поиска по сетке из вышеперечисленных значений параметров. На основе этого, по усредненному значению в кросс-валидации, выбирался лучший набор и после его определения строилась финальная модель, которая дает 87,5% точности на кросс-валидации и 86.5% точности на отложенной выборке. В рамках данной задачи такой точности может быть вполне достаточно для подведения промежуточных результатов. Стоит дополнительно отметить, что помимо данного алгоритма, были построены более сильные с точностью в 93% (XGBoost) и что качество модели деревьев решений можно в дальнейшем улучшать благодаря поиску новых признаков (feature engineering), подбору более широкого спектра значений параметров, оптимизации величины зазора между данными с детектора и модельными данными на основе важности признаков. Относительно последнего, стоит отметить, что деревья решений могут предоставить информацию о важности признаков (*рисунок 16*) в процессе построения, то есть сказать, насколько сильно тот или иной признак влияет на отбор при построении дерева.

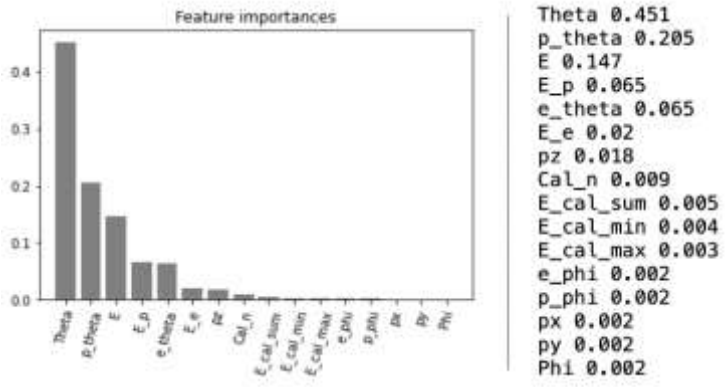

Рисунок 16 Важность признаков при построении модели деревьев решений. Значения указаны в долях

Результаты

Полученные результаты могут послужить базой для совершенствования критериев отбора. Для этого необходимо провести физический анализ полученных результатов.

Далее предоставлены два формата финального дерева, в виде блок-схемы и в виде схемы по отбору на распределениях (*рисунок 17* и *18*). Стоит отметить, что дерево получилось достаточно глубоким (финальная глубина составляет 10 уровней), и его полное отображение весьма затруднительно, в связи с этим будут представлена лишь его верхняя часть.

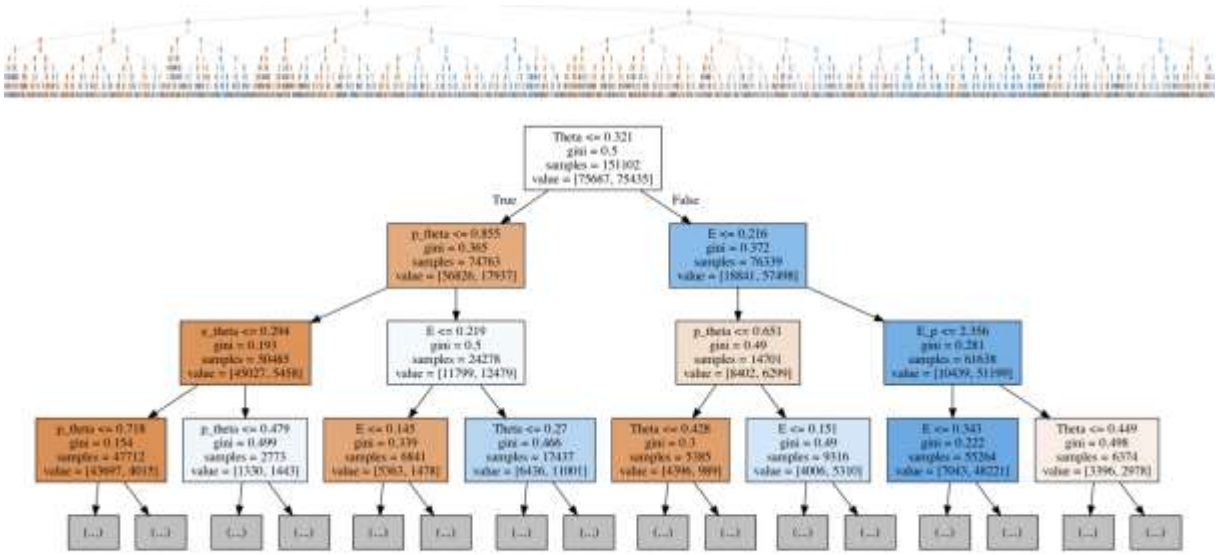

Рисунок 17 Первые 4 уровня дерева решений в виде блок-схемы. Полный вид дерева представлен в верхней части рисунка.

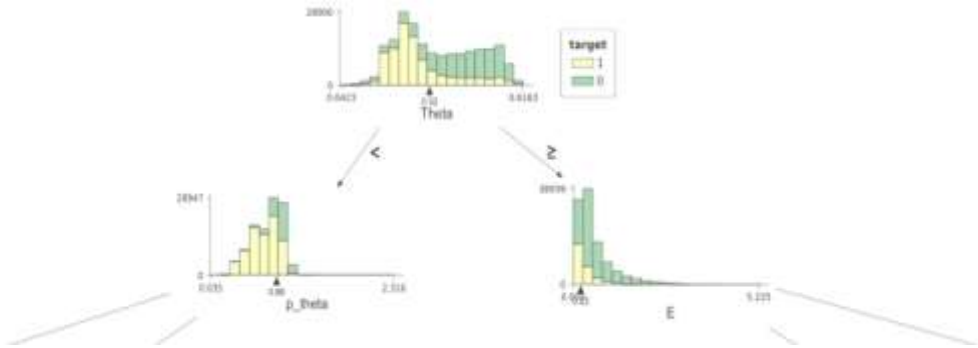

Рисунок 18 Первые 2 уровня дерева решений в виде схемы распределений. Здесь наглядно видно по каким порогам, относительно полной выборки, дерево делает разделение.

В качестве результатов данной работы можно предоставить сравнение двух гистограмм (*рисунок 19*). Обе гистограммы строились для инвариантной массы двух гамма-квантов. Для первой гистограммы были взяты экспериментальные данные детектора CLAS12. В конечном состоянии требовалось наличие только 1 электрона, 1 протона и 2 гамма-квантов. Для второй были взяты данные детектора с условием, что в реакции

присутствует только 1 электрон и протон, и как минимум 2 гамма-кванта, после чего эти данные были пропущены через модель дерева решений и аналогично построено распределение инвариантной массы.

По данным гистограммам видно, что алгоритм работает корректно, сильно подавляет фон и сохраняет Гауссову форму пика. В результате проделанной работы мы имеем набор алгоритмов, которые имеют возможность отсеивать фоновые реакции, по которым также можно усовершенствовать кинематические отборы, либо просто пользоваться уже отобранными событиями.

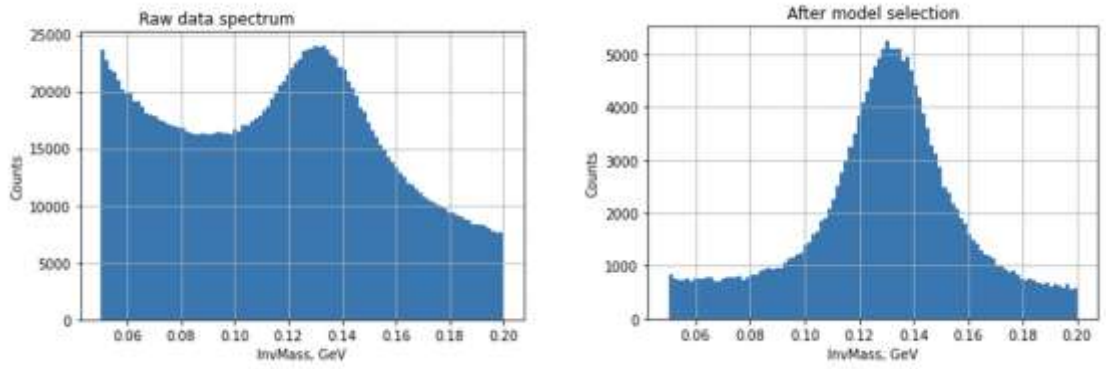

Рисунок 19 Гистограммы для инвариантной массы. Слева по данным без отбора, справа по данным после работы алгоритма.

Заключение

В рамках данной работы были написаны программные блоки для отбора данных, которые использовались для дальнейшего обучения, проведена процедура их предварительной подготовки и сформированы финальные размеченные наборы данных. Построены различные модели отбора гамма-квантов от распада нейтрального пи-мезона на основе алгоритмов машинного обучения, проведена процедура их независимой настройки, измерена их точность, а результаты конечной модели визуализированы в виде блок-схемы и схемы по распределениям. Проведен отбор гамма-квантов от искомых реакций ($\boldsymbol{e^- + p \rightarrow e^- + p + \pi^0 \rightarrow e^- + p + 2\gamma}$) при

помощи финальной модели деревьев решений и построены распределения инвариантной массы отобранных гамма-квантов.

Список литературы

[1] - V.D. Burkert, Jefferson lab at 12 GeV: The science program, Ann. Rev. Nucl. Part. Sci. 68 (2018) 405, http://dx.doi.org/10.1146/annurev-nucl-101917-021129.

[2] - Jlab.org «An Accelerator overview» https://www.jlab.org/accelerator/ops-orientation/acc_overview.

[3] - V.D.Burkert, L.Elouadrhiri «The CLAS12 spectrometer at Jefferson Laboratory» Nuclear Inst. And Methods in Physics Research, A (2018)3.

[4] - M. Amarian, et al., The CLAS forward electromagnetic calorimeter, Nucl. Instrum. Methods A 460 (2001) 239, http://dx.doi.org/10.1016/S0168-9002(00)00996-7.

[5] - Tom M. Mitchell «Machine learning» https://www.cin.ufpe.br/~cavmj/Machine - Learning - Tom Mitchell.pdf (1997) 2

[6] - Open Data Science «Открытый курс машинного обучения» https://habr.com/en/company/ods/blog/322534/, (2017) 3

[7] - «Top 10 algorithms in data mining», Knowledge and Information Systems, http://www.ar.sanken.osaka-u.ac.jp/~motoda/papers/10Algorithms.pdf (2008) 3

[8] - T. Chen, C. Guestrin «XGBoost: A Scalable Tree Boosting System», https://arxiv.org/pdf/1603.02754.pdf (2016) 2

[9] - GitHub repository https://github.com/GOLoDovkA-A/Class12Diploma

[10] - Scikit-learn main page https://scikit-learn.org/stable/

[11] - Scikit-learn «Decision Tree Classifier» https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[12] - GitHub repository «JeffersonLab» https://github.com/JeffersonLab/clas12-mcgen

[13] - Е.Соколов «Семинары по метрическим методам классификации» (2013) http://www.machinelearning.ru/wiki/images/9/9a/Sem1_knn.pdf

[14] - О. В. Чумак «Энтропии и фракталы в анализе данных»
http://www.sai.msu.ru/amateur/books/Maket_Chumak_1.pdf (2011) 19,65

[15] - Open Data Science «Открытый курс машинного обучения»
https://habr.com/ru/company/ods/blog/324402/