**АННОТАЦИЯ**

На основании задания на дипломную работу разработан проект программного комплекса интеллектуального анализа игровой статистики и частично реализованы его компоненты, позволяющего прогнозировать исход матча между двумя командами киберспортсменов в игре *Dota 2* на основе скоринговой модели.

В аналитической части рассмотрен процесс прогнозирования исхода матча между двумя командами в игре *Dota 2*. Произведен сравнительный анализ некоторых существующих программных аналогов, отмечены их достоинства и недостатки, а также сделан вывод о необходимости создания программного комплекса.

В техническом задании описаны основания для разработки, назначение разработки, основные требования к программному комплексу.

На этапе проектирования разработаны и описаны архитектура программной системы, функциональная модель, модель данных и структурное проектирование программного комплекса.

На этапе тестирования проведены и описаны тестирование программного комплекса на соответствие функциональным требованиям, нагрузочное тестирование сервиса прогнозирования исходов матчей, тестирование верстки, сравнение моделей с разным числом входных факторов.

В экономической части произведены расчеты затрат на создание программного комплекса.

В организационной части даны общие рекомендации по охране труда и технике безопасности.

СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 3](#_Toc12201654)

[ВВЕДЕНИЕ 7](#_Toc12201655)

[1. АНАЛИЗ ТРЕБОВАНИЙ 9](#_Toc12201656)

[1.1. Описание игры Dota 2 9](#_Toc12201661)

[1.2. Описание процесса прогнозирования исхода матча между двумя командами в игре Dota 2 10](#_Toc12201662)

[1.3. Обзор и сравнение аналогов 11](#_Toc12201663)

[1.3.1. Dotabuff.com 11](#_Toc12201668)

[1.3.2. Dota Plus 13](#_Toc12201669)

[1.3.3. Dotapicker.com 14](#_Toc12201670)

[1.3.4. Прогнозирующие модели 15](#_Toc12201671)

[1.4. Результаты сравнительного анализа 16](#_Toc12201672)

[2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ 18](#_Toc12201673)

[2.1. Основания для разработки 18](#_Toc12201674)

[2.2. Назначение разработки 18](#_Toc12201675)

[2.3. Функциональные требования 18](#_Toc12201676)

[2.3.1. Общие требования 18](#_Toc12201677)

[2.3.2. Требования к подсистеме авторизации 18](#_Toc12201678)

[2.3.3. Требования  к  подсистеме  загрузки  и  парсинга  реплеев матчей 19](#_Toc12201679)

[2.3.4. Требования к подсистеме скорингового анализа 19](#_Toc12201680)

[2.3.5. Требования к подсистеме персональной статистики 21](#_Toc12201681)

[2.4. Требования к пользовательскому интерфейсу 22](#_Toc12201682)

[2.5. Требования к верстке сайта 22](#_Toc12201683)

[2.6. Требования к надежности 23](#_Toc12201684)

[2.7. Требования к техническим и программным средствам 24](#_Toc12201685)

[2.8. Условия эксплуатации 24](#_Toc12201686)

[3. ЭКОНОМИЧЕСКИЙ АНАЛИЗ 25](#_Toc12201687)

[3.1. Организационная структура проекта 25](#_Toc12201690)

[3.2. Календарный план проекта 25](#_Toc12201691)

[3.3. Расчет затрат на разработку продукта 29](#_Toc12201692)

[3.3.1. Расчет заработной платы исполнителей работ по созданию проекта 29](#_Toc12201693)

[3.3.2. Расчет отчислений на социальные нужды 30](#_Toc12201694)

[3.3.3. Арендные платежи за офисные помещения 31](#_Toc12201695)

[3.3.4. Амортизация используемых основных средств и нематериальных активов 31](#_Toc12201696)

[3.3.5. Расходы на модернизацию и приобретение основных средств 31](#_Toc12201697)

[3.3.6. Расходы на приобретение ПО 32](#_Toc12201698)

[3.3.7. Расходы на интернет и связь 32](#_Toc12201699)

[3.3.8. Расходы на канцелярские товары и расходные материалы 32](#_Toc12201700)

[3.3.9. Прочие расходы 32](#_Toc12201701)

[3.3.10. Расчет себестоимости проекта 32](#_Toc12201702)

[4. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ 34](#_Toc12201703)

[4.1. Цель и задачи исследования 34](#_Toc12201708)

[4.2. Описание процесса скорингового анализа 35](#_Toc12201709)

[4.3. Описание модели логистической регрессии 40](#_Toc12201710)

[4.4. Определение независимых и зависимых признаков 42](#_Toc12201711)

[4.5. Сбор и подготовка данных 46](#_Toc12201712)

[4.6. Оценка качества и очистка данных 48](#_Toc12201713)

[4.7. Сэмплинг 51](#_Toc12201714)

[4.8. Двумерный анализ признаков 51](#_Toc12201715)

[4.9. Сокращение числа признаков 55](#_Toc12201716)

[4.10. Разбиение на обучающее и тестовое множества 57](#_Toc12201717)

[4.11. Оценка качества и интерпретация модели 59](#_Toc12201718)

[4.11.1. ROC-анализ 60](#_Toc12201732)

[4.11.2. Проверка значимости и интерпретация модели 65](#_Toc12201733)

[4.11.3. Оценка качества скоринговой карты 70](#_Toc12201734)

[4.12. Результаты и рекомендации по улучшению модели 74](#_Toc12201735)

[5. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА 77](#_Toc12201738)

[5.1. Функциональная модель 77](#_Toc12201739)

[5.2. Архитектура программной системы 78](#_Toc12201740)

[5.3. Средства разработки 80](#_Toc12201741)

[5.4. Модель данных 81](#_Toc12201742)

[5.5. Структурное проектирование 83](#_Toc12201743)

[6. ТЕСТИРОВАНИЕ ПРОГРАММНОГО КОМПЛЕКСА 88](#_Toc12201744)

[6.1. План испытаний 88](#_Toc12201747)

[6.2. Функциональное тестирование 88](#_Toc12201748)

[6.2.1. Тестирование авторизации 88](#_Toc12201749)

[6.2.2. Тестирование сервиса прогнозирования исходов матчей 89](#_Toc12201750)

[6.2.3. Тестирование анализа «Что-если» 89](#_Toc12201751)

[6.3. Тестирование скоринговой модели 91](#_Toc12201752)

[6.3.1. Сравнение моделей с разным числом входных факторов 91](#_Toc12201753)

[6.4. Нагрузочное тестирование сервиса прогнозирования исходов матчей 92](#_Toc12201754)

[6.5. Тестирование верстки 93](#_Toc12201755)

[7. ОРГАНИЗАЦИОННАЯ ЧАСТЬ 94](#_Toc12201756)

[7.1. Организация режима труда и отдыха 94](#_Toc12201757)

[7.2. Электробезопасность 95](#_Toc12201758)

[7.3. Синдром запястного канала 96](#_Toc12201759)

[7.4. Требования к освещенности 97](#_Toc12201760)

[7.5. Пожарная безопасность 98](#_Toc12201761)

[7.6. Расчет освещенности 100](#_Toc12201762)

[ЗАКЛЮЧЕНИЕ 104](#_Toc12201763)

[СПИСОК ЛИТЕРАТУРЫ 106](#_Toc12201764)

[ПРИЛОЖЕНИЕ А 111](#_Toc12201765)

# ВВЕДЕНИЕ

На данный момент индустрия компьютерных игр одна из самых быстрорастущих в мире. Неудивительно, что через нее проходят огромные денежные потоки. Например, призовой фонд турнира *The International* *2018* по компьютерной многопользовательской командной игре *Dota 2* составил 25 миллионов долларов [1]. Далеко не каждый вид спорта может похвастаться такими суммами. В марте этого года максимальный онлайн в *Dota 2* составил чуть больше 1 миллиона игроков, и это не самый высокий показатель за все время существования игры [2]. В разных странах поднимается вопрос о признании киберспорта официальным видом спорта. В России компьютерный спорт включен в реестр официальных видов спорта в 2016 году [3]. В Китае министерство трудовых ресурсов и социального обеспечения официально утвердило две профессии: «киберспортсмен» и «киберспортивный менеджер» [4].

Очевидно, что теперь для многих людей компьютерные игры не просто развлечение, а способ зарабатывания денежных средств: организация и участие в турнирах, ставки на киберспортивные события, продажа внутриигровых виртуальных предметов, стриминг и другое.

Для того чтобы помочь новичкам и опытным игрокам отслеживать свои результаты и повышать свой уровень игры в *Dota 2*, существуют программные продукты, которые предоставляют как персональную статистику для отдельных игроков и команд, так и разного рода общую статистику по игре, собранную на основе миллионов общедоступных матчей. Поскольку имеется свободный доступ к такому количеству игровых данных, значит можно попробовать разработать статистическую модель для прогнозирования исхода матча между двумя командами в игре *Dota 2*, которая ляжет в основу будущего программного комплекса. Во-первых, сервис по прогнозированию результатов матчей пользовался бы спросом у людей, делающих ставки на киберспортивные события. Во-вторых, у игроков, участвующих во внутриигровых квестах по предсказанию результатов матчей в турнирах, за которые они получают бонусы в игре.

С помощью разрабатываемого программного комплекса, а также опираясь на мнения профессиональных игровых аналитиков и собственные суждения, лицо, принимающее решение, сможет сделать более взвешенный выбор в ставках на киберспортивные матчи по игре *Dota 2*. Также игрокам будет доступна персональная статистика, чтобы они могли отслеживать свои результаты в *Dota 2*.

**Цель данной работы –** разработка проекта программного комплекса интеллектуального анализа игровой статистики и частичная реализация подсистемы скорингового анализа.

**Задачи работы:**

1. изучить предметную область;
2. проанализировать существующие аналоги;
3. разработать скоринговую модель;
4. спроектировать программный комплекс;
5. разработать подсистему скорингового анализа;
6. протестировать на соответствие функциональным требованиям;
7. проанализировать полученные результаты;
8. предложить план дальнейшего развития и модернизации программного комплекса.

**Объект исследования –** процесс прогнозирования исхода матча между двумя командами в игре *Dota 2*.

**Предмет исследования –** автоматизация процесса прогнозирования исхода матча между двумя командами в игре *Dota 2*.

# 1. АНАЛИЗ ТРЕБОВАНИЙ

Далее приводится описание игры *Dota 2*, предметной области и сравнительный анализ существующих программных аналогов, для того чтобы изучить различные варианты реализации инструментов, имеющихся у аналогичных систем, и определить требования к характеристикам разрабатываемого программного комплекса.



## 1.1. Описание игры Dota 2

*Dota 2* – компьютерная многопользовательская командная игра, разработанная компанией *Valve Corporation*. Команда *Radiant* (силы света) сражается с командой *Dire* (силы тьмы). В команде 5 игроков, каждый из которых управляет одним героем с определенным набором способностей. Перед началом матча игроки выбирают героев из более 100 возможных, все 10 героев обязательно различны. По ходу матча герой получает очки опыта, зарабатывает золото, например, за убийство вражеского героя, покупает и собирает предметы, которые делают его сильнее. Матч проходит на карте квадратной формы. В левом нижнем углу карты располагается база команды сил света, а в правом верхнем – база сил тьмы. Цель игры – разрушить трон, находящийся на базе противника.

На сегодняшний день *Dota 2* является активной киберспортивной дисциплиной, в которой профессиональные команды со всего мира соревнуются в различных турнирах. Ежегодный турнир с самым большим призовым фондом – *The International*. Начиная с 2017 года компанией *Valve Corporation* и ее партнерами организуютсяпрофессиональные сезоны *Dota Pro Circuit*, состоящие из нескольких турниров, за призовые места в которых команды получают баллы [5]. Несколько команд, набравших наибольшее количество очков за сезон, получают прямые приглашения на *The International*.

В интернете имеется множество сайтов, которые предоставляют различную статистику по игре *Dota 2*, собранную на основе миллионов общедоступных матчей. Например, *Dotabuff.com* [6] и *Opendota.com* [7]. В *Dota 2* существует общий одиночный рейтинг игроков, который можно рассматривать в качества показателя, отражающего уровень мастерства игрока [8]. За каждую победу рейтинг увеличивается на определенное количество очков, за поражение – уменьшается.



Рис. 1.1. Игровой процесс в Dota 2

## 1.2. Описание процесса прогнозирования исхода матча между двумя командами в игре Dota 2

Особый интерес представляет прогнозирование, какая из команд одержит победу в конкретном матче. Вручную это делать тяжело, поскольку матчей играется много, а прогноз может основываться на множестве факторов: результаты команд по итогам последних турниров, рейтинг игроков, стиль игры, выбранные герои в матче и т.д. К тому же у каждого игрока свое мнение насчет того, какие факторы считать более или менее важными. Поэтому для успешных прогнозов необходимо постоянно следить за турнирами, анализировать статистику команд на специализированных сайтах. В большинстве случаев этот процесс занимает много времени.

Для решения данной проблемы необходимо разработать программный комплекс, который будет автоматизировать процесс прогнозирования исхода матча между двумя командами. Игра *Dota 2* в достаточной мере детерминирована, т.е. умение игроков, стратегия в зависимости от выбранных героев и как игроки действуют по ходу игры в большей степени влияют на результат, чем фактор случайности. Значит можно попробовать применить алгоритмы машинного обучения для разработки прогнозирующей модели, которая станет основой для сервиса прогнозирования исходов матчей между двумя командами в игре *Dota 2*.

## 1.3. Обзор и сравнение аналогов



### 1.3.1. Dotabuff.com

*Dotabuff* [6] – веб-ресурс, где миллионы игроков *Dota 2* получают подробную статистику и лучше понимают игру. *Dotabuff* является одним из лидирующих ресурсов *Dota 2* сообщества как для новичков, так и для опытных игроков.

Данный веб-ресурс предоставляет следующую персональную статистику для игроков:

* статистика по добыче золота в матче (общая ценность, количество заработанного золота, зарабатываемое золото и получаемый опыт в минуту, количество добитых крипов, золото от убийств и смертей);
* статистика по обзору на карте в матче (карта обзора, количество приобретенных, поставленных и разбитых вардов, купленных и использованных предметов *Dust*, *Smoke* и *Gem*);
* подробный журнал матча с фильтрами, который показывает, когда и какие события происходили по ходу игры;
* временная шкала покупки предметов и изучения способностей героев (рис. 1.2);

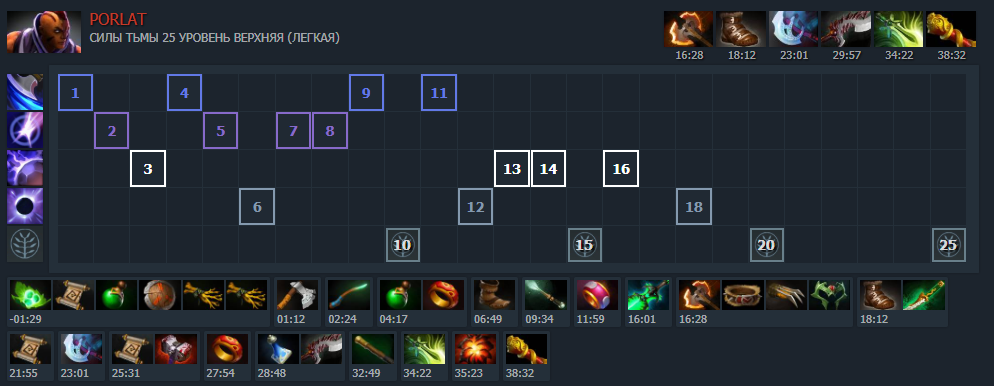


Рис. 1.2. Временная шкала покупки предметов и изучения способностей героев

* таблица рекордов (самый длинный матч, наибольшее число убийств, помощи, урона по строениям, соучастий в убийствах, длиннейшая серия побед и поражений и т.д.) (рис. 1.3);
* и многое другое.



Рис. 1.3. Таблица рекордов

Также веб-ресурс предоставляет подробную статистику по каждой команде, в частности:

* общая информация (дата основания, доля побед, список игроков, список матчей, участие в турнирах, рейтинг *Dota Pro Circuit*);
* статистика по каждому герою (количество матчей, доля побед, среднее количество золота и опыта, среднее число убийств крипов, список матчей, показатель *KDA*, статистика против героев соперников);
* статистика по каждому игроку (количество матчей, доля побед, среднее количество золота и опыта, среднее число убийств крипов, список матчей, показатель *KDA*, статистика по героям).

Достоинства: большое количество детализированной статистики в целом, возможность сортировки и фильтрации данных по множеству критериев, красочность и понятность графиков.

Данный веб-ресурс не имеет функции предсказания исходов матчей. Но с помощью него можно бесплатно сравнить статистику двух команд, только для этого необходимо совершить множество переходов по страницам сайта, применяя фильтрацию данных, поскольку разная статистика находится на отдельных страницах сайта, т.е. оценивать шансы придется вручную.

### 1.3.2. Dota Plus

*Dota Plus* [9] – это сервис по подписке, встроенный в игру *Dota 2*. Став подписчиком *DotaPlus* за 4 доллара в месяц, игрок получает дополнительные функции в самой игре. Одной из таких функций является график вероятности победы, который рассчитывается на основе текущей игровой ситуации в матче (рис 1.4). График доступен сразу после начала матча, но поскольку до этого момента никакие события в игре не происходили, можно предположить, что изначально прогноз основан только на информации о выбранных командами героях. Вероятности можно посмотреть для всех матчей, даже в которых играют одиночные игроки. К тому же было замечено, что, когда в матче явно имеется команда-фаворит, шансы на победу у нее могут быть меньше в начале игры. Это свидетельствует о том, что, скорее всего, в самом начале матча учитываются герои и не учитывается статистика команд.

К плюсам можно отнести доступность прогноза на протяжении всего матча, к тому же данный инструмент от самих разработчиков игры *Dota 2*.

К недостаткам можно отнести доступность только в самой игре, отсутствие точной информации о том, на каких данных основан прогноз, отсутствие сравнительной статистики команд, стоимость 4 доллара в месяц.



Рис. 1.4. График вероятности победы в Dota 2

### 1.3.3. Dotapicker.com

*Dotapicker* [10] – веб-ресурс, который помогает игрокам выбирать синергирующих между собой и более эффективных героев в *Dota 2* против героев соперников, чтобы увеличить вероятность победы. Используя данный сервис, можно примерно оценить шансы команд на победу на основе коэффициентов эффективности героев друг против друга и синергии их между собой, которые статистически рассчитываются на основе данных из миллионов матчей (рис. 1.5). Чем больше значения коэффициентов, тем выше шансы на победу.

Среди плюсов можно выделить, что данный сервис бесплатный, а также отметить, что только на основе информации об эффективности героев друг против друга и синергии между собой можно разработать статистическую модель с точностью прогнозирования от 70% и выше, о чем будет сказано далее в данной работе.

Среди недостатков можно выделить трудность интерпретации коэффициентов эффективности и синергии с точки зрения вероятностей и отсутствие других критериев сравнения.

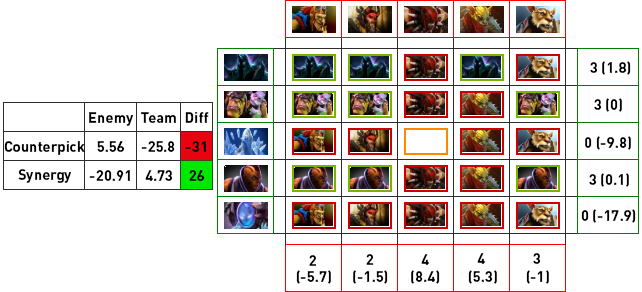


Рис. 1.5. Эффективность и синергия героев

### 1.3.4. Прогнозирующие модели

В статье *“How Does He Saw Me? A Recommendation Engine for Picking Heroes in Dota 2”* авторы описывают модели на основе логистической регрессии и метода *k*-ближайших соседей [11]. В первом случае в качестве входных факторов выступает информация только о выбранных героях командами в матче, точность прогноза составила 69,8%, для обучения использовались данные из 18000 матчей. В втором случае учитывалось то, как герои сочетаются друг с другом, доля верно классифицированных примеров составила 70% для выборки из 50000 матчей.

В статье *“Result Prediction by Mining Replays in Dota 2”* авторы описывают модель на основе алгоритма «случайного леса» (*Random forest*) [12]. Для обучения и тестирования модели использовались данные из 15146 матчей, а в качестве входных факторов – данные по каждому герою по ходу матча (золото в минуту, количество убийств, смертей и т.д.). Точность модели на основе данных за первые 5 минут матча составила 82,23%. Несмотря на такую высокую точность, данную модель можно использовать для предсказания только через определенное время после начала матча.

В статье *“To win or not to win? A prediction model to determine the outcome of a DotA2 match”* автор описывает модель на основе алгоритмов логистической регрессии и пары из логистической регрессии и генетического алгоритма [13]. В качестве входных факторов в первом случае используется информация только о выбранных героях командами в матче, во втором случае также учитываются комбинации героев. Для обучения и тестирования модели использовались данные из 18500 матчей. В первом случае точность составила 69,42%, во втором – 74,1%.

Следует отметить, что в приведенных выше статьях авторы использовали различные подходы для определения того, насколько одни герои сочетаются и не сочетаются с другими, одни герои слабее и сильнее других. В зависимости от выбора входных факторов, модели могут применяться как для прогнозирования результатов матчей в режиме реального времени, так и до их начала.

В других похожих статьях [14, 15, 16] в большинстве случаев использовался алгоритм логистической регрессии, точность моделей варьировалась от 55% до 80%.

## 1.4. Результаты сравнительного анализа

На основании сравнительного анализа можно сделать вывод, что в целом рассмотренные решения можно рассматривать в качестве помощника для предсказания результатов матчей, но в большинстве случаев придется самостоятельно оценивать шансы на победу для команд, анализируя имеющиеся данные. В каждом конкретном случае применяются различные подходы к оценке и имеются свои недостатки. Также автор ознакомился с различными подходами и алгоритмами, описанными в статьях, которые могут применяться при реализации модели, прогнозирующей исход матча по игре *Dota 2*. Но среднестатистический пользователь, незнакомый с алгоритмами машинного обучения и программированием, не сможет использовать эту информацию в своих целях.

Таким образом, было принято решение разработать статистическую модель, которая легла в основу разрабатываемого программного комплекса, позволяющего прогнозировать вероятность победы одной из команд в матче с отображением сравнительной статистики команд, для того чтобы было понятно, на основе каких данных получен тот или иной результат. Также игрокам будет доступна персональная статистика, чтобы они могли отслеживать свои результаты и повышать свой уровень игры в *Dota 2*.

**Цель данной работы –** разработка проекта программного комплекса интеллектуального анализа игровой статистики и частичная реализация подсистемы скорингового анализа.

**Задачи работы:**

1. изучить предметную область;
2. проанализировать существующие аналоги;
3. разработать скоринговую модель;
4. спроектировать программный комплекс;
5. разработать подсистему скорингового анализа;
6. протестировать на соответствие функциональным требованиям;
7. проанализировать полученные результаты;
8. предложить план дальнейшего развития и модернизации программного комплекса.

**Объект исследования –** процесс прогнозирования исхода матча между двумя командами в игре *Dota 2*.

**Предмет исследования –** автоматизация процесса прогнозирования исхода матча между двумя командами в игре *Dota 2*.

# 2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

## 2.1. Основания для разработки

Основанием для разработки программного комплекса интеллектуального анализа игровой статистики является задание, выданное к.т.н., доц. Лагеревым Д.Г. на основании приказа по БГТУ «О допуске к выполнению выпускных квалификационных работ, утверждении тем выпускных квалификационных работ и руководителей» №420-3 от 17.05.2019.

## 2.2. Назначение разработки

Разрабатываемый программный комплекс осуществляет прогнозирование исходов матчей между двумя командами в игре *Dota 2* на основе статистической модели и отображает сравнительную статистику двух команд для конкретного матча. Данный сервис предназначен для людей, делающих ставки на киберспортивные события, и игроков, участвующих во внутриигровых квестах по предсказанию результатов матчей, за которые они получают бонусы в игре *Dota 2*. Другой важной функцией является предоставление персональной статистики для игроков *Dota 2*, для того чтобы они могли отслеживать свои результаты и повышать свой уровень игры.

## 2.3. Функциональные требования

### 2.3.1. Общие требования

Программный комплекс должен быть реализован в виде веб-приложения. Предлагается выделить следующие функциональные подсистемы: подсистема скорингового анализа (подсистема прогнозирования результатов матчей), подсистема персональной статистики, подсистема загрузки и парсинга реплеев матчей, подсистема авторизации, подсистема работы с БД.

### 2.3.2. Требования к подсистеме авторизации

Поскольку аккаунт *Dota 2* жестко привязан к игровой платформе *Steam*,

авторизация должна происходить через данную платформу с помощью механизма *Steam OpenID*. Если пользователь авторизуется на сайте первый раз, то логин и 17-значный идентификатор его *Steam*-аккаунта сохраняются в базе данных, т.е. осуществляется регистрация пользователя в системе.

### 2.3.3. Требования к подсистеме загрузки и парсинга реплеев матчей

Необходимые игровые данные должны извлекаться из файлов реплеев матчей, поэтому необходимо разработать программу-парсер или использовать любое готовое решение, имеющееся в открытом доступе, при необходимости доработать. Обработка одного файла должна занимать не более 5 секунд. Файлы реплеев матчей должны загружаться с серверов игры *Dota 2* посредством *Steam Web* *API*.

### 2.3.4. Требования к подсистеме скорингового анализа

Прогнозирование результатов матчей должно осуществляться на основе скоринговой модели. Используемый алгоритм – множественная логистическая регрессия. Прогностическая точность модели или доля верно классифицированных примеров должна составлять не менее 65% на обучающем и тестовом множествах. Помимо реплеев матчей, необходимые статистические данные для построения модели рекомендуется брать с веб-ресурсов *Dotabuff.com* [6] и *Opendota.com* [7].

Взаимодействие пользователя с данной подсистемой должно осуществляться следующим образом:

1) выбираются две команды, участвующие в матче, из списка доступных команд;

2) выбираются 10 героев, выбранных командами;

3) производится клик мышью на кнопку запуска прогнозирования (происходит запрос на сервер);

4) отображаются вероятности побед для команд в процентах, их сравнительная статистика и скоринговые баллы (пришел ответ от сервера).

Должна отображаться следующая сравнительная статистика команд:

* доля побед;
* доля побед команд друг против друг и результаты последних 10 матчей;
* доля матчей с числом смертей меньше 25;
* средняя продолжительность матчей;
* таблица эффективности героев команды сил света против героев команды сил тьмы;
* доля побед команды за выбранных героев и против героев соперников.

Должны быть реализованы следующие функции, доступ к которым имеет только администратор:

* обновление данных, на которых обучается и тестируется модель;
* сэмплинг – разбиение всех данных на обучающее и тестовое множества (перед запуском необходимо задать размеры множеств, выбрать метод сэмплинга: случайный, равномерный случайный, стратифицированный, последовательный);
* обучение модели (перед запуском необходимо задать максимальное число итераций алгоритма расчета коэффициентов, точность функции оценки, порог отсечения);
* расчет скоринговой карты (перед запуском необходимо задать параметры масштабирования карты);
* анализ «Что-если», который позволяет рассчитывать вероятности и скоринговые баллы для заданных значений входных факторов;
* просмотр отчета по модели с возможностью скачивания (формат файла – *“*.*xlsx”*), который включает в себя значения рассчитанных коэффициентов, критериев значимости, данные о качестве классификации, скоринговую карту.

### 2.3.5. Требования к подсистеме персональной статистики

Пользователю должна быть доступна следующая игровая статистика:

* статистика по добыче золота в матче (общая ценность, количество заработанного золота, зарабатываемое золото и получаемый опыт в минуту, количество добитых крипов, золото от убийств и смертей);
* статистика по рунам в матче (журнал рун, количество и места появления рун, какой герой и когда активировал руну);
* временная шкала покупки предметов и изучения способностей героев;
* статистика целей в матче (количество разрушенных башен и зданий, урон по строениям, статистика по *Aegis*, *Cheese*, рунам, журнал целей);
* экономика убийств и смертей в матче (карта смертей, журнал смертей, количество убийств и смертей, помощи, выкупов, потерянного и заработанного золота);
* статистика по каждому герою за все время (процент побед и поражений, количество сыгранных матчей, количество убийств, смертей и помощи, добыча золота и опыта в минуту, роли);
* статистика сражений в матче (нанесенный и полученный урон, полученное и использованное лечение, время бездействия героя, временная шкала нанесенного урона);
* графики общей ценности отдельных игроков и команды в целом, добычи золота и опыта;
* таблица рекордов за все время (самый длинный матч, наибольшее количество убийств, смертей, помощи, добитых крипов, золота, опыта, урона по героям, лечения героев, урона по строениям, соучастий в убийствах, длиннейшая серия побед и поражений);
* общая статистика за все время (количество сыгранных матчей, доля побед и поражений, режимы игры, стороны, регионы);
* статистика по предметам за все время (самые используемые, доля побед, вклад в игру, количество матчей).

Пользователь должен иметь возможность запрашивать немедленное обновление данных (нажатием на определенную кнопку на веб-странице) и настраивать частоту автоматического обновления данных (каждые *n*-дней). Если в течение заданного периода пользователь не будет активен (т.е. не сыграет ни одного матча), то автоматическое обновление статистики для него приостанавливается. Для возобновления необходимо запустить немедленное обновление данных, тогда пользователь снова будет считаться активным.

В настройках профиля пользователь может задать, будут ли видны его результаты остальным игрокам.

## 2.4. Требования к пользовательскому интерфейсу

Все статистические данные следует отображать в виде таблиц, диаграмм и графиков. При разработке интерфейса рекомендуется ориентироваться на существующие аналогичные системы: *Dotabuff.com* [6] и *Opendota.com* [7], использовать терминологию и изображения из игры *Dota 2*.

Должна присутствовать возможность сортировки табличных данных по столбцам.

## 2.5. Требования к верстке сайта

Верстка страниц сайта в формате языка гипертекстовой разметки (*HTML*), включая каскадные таблицы стилей (*CSS*) и необходимые скрипты на языке программирования *JavaScript*, должна соответствовать следующим требованиям:

* изменения *hover*-состояний и *focus*-эффекты полей ввода и кнопок сопровождаются микроанимациями и *transition*-эффектами;
* обязательно комментирование крупных смысловых блоков в целях читаемости кода;
* отсутствие горизонтальной полосы прокрутки страницы на мониторах с разрешением 1024 пикселей в ширину и выше;
* валидация форм ввода данных;
* корректное отображение в браузерах *Google Chrome* версии 21 и выше, *Mozilla Firefox* версии 28 и выше, *Opera* версии 15 и выше.

## 2.6. Требования к надежности

Разрабатываемый программный комплекс должен удовлетворять следующим требованиям надежности:

* программная и аппаратная часть должны быть проработаны на наличие узких мест, способных привести к сбою;
* отказы в работе программы и порча данных вследствие некорректных действий пользователя недопустимы;
* сайт должен быть устойчив к наиболее частым методам взлома, и необходимо свести к минимуму возможность более редких;
* время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств;
* время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Заказчиком должно быть организовано бесперебойное питание технических средств и использоваться лицензионное программное обеспечение.

## 2.7. Требования к техническим и программным средствам

Конфигурация компьютера, на котором будет располагаться сервер, должна иметь следующий минимальный набор характеристик и программных средств:

* процессор *Intel/AMD* с тактовой частотой не менее 2 ГГц;
* оперативная память не менее 8 Гб;
* жесткий диск с не менее чем 512 Гб свободного пространства;
* операционная система *Windows* *Server 2008 R2* или новее;
* программный пакет языка программирования *PHP* версии 7.2 или выше;
* СУДБ *MySQL* версии 5.8 или выше;
* *Apache HTTP*-сервер версии 2.4 или выше;
* программное обеспечение *Deductor Studio Academic* версии 5.3.

## 2.8. Условия эксплуатации

Условия эксплуатации, необходимые для корректной работы компьютерного оборудования и разрабатываемого продукта:

* должна быть обеспечена защита от электромагнитных и ионизирующих излучений;
* эксплуатационные режимы работы не должны превышать значений, указанных в технической характеристике оборудования;
* для нормального функционирования компьютерного комплекса прочие условия должны соответствовать санитарным нормам и правилам эксплуатации компьютеров;
* минимальное количество персонала, требуемого для работы программы – 2 (пользователь и системный администратор).

# 4. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ



## 4.1. Цель и задачи исследования

Было принято решение реализовать подсистему для оценки вероятности выигрыша одной из команд киберспортсменов в компьютерной игре *Dota 2* на основе скоринговой модели, используя для ее разработки статистический метод – логистическую регрессию.

Поскольку эта первое серьезное знакомство автора с машинным обучением, а логистическая регрессия является наиболее часто используемой на практике математической моделью для построения скоринговых карт [17], к тому же, как было отмечено ранее, она нередко используется для разработки прогнозирующих моделей, связанных с игрой *Dota 2*, был сделан выбор в пользу этого алгоритма. В свою очередь, скоринговые модели достаточно просты для понимания и популярны, используются в банках, страховых, маркетинговых компаниях и в других сферах бизнеса, по разработке скоринговых систем имеется множество литературы.

Следовательно, цель – разработать скоринговую модель, которая на вход принимает определенные статистические данные о командах, игроках, героях, а на выходе выдает вероятность победы для каждой команды и скоринговые баллы.

Для того чтобы разработать скоринговую модель, необходимо выполнить следующие задачи:

* собрать и подготовить, оценить качество и при необходимости произвести очистку данных;
* выполнить сэмплинг;
* отобрать наиболее значимые входные факторы;
* обучить и протестировать модель;
* рассчитать скоринговую карту;
* оценить качество и проверить значимость скоринговой модели.

## 4.2. Описание процесса скорингового анализа

Скоринг (от англ. *scoring* – подсчет очков в игре) – процедура ранжирования объектов в соответствии с их измеренными характеристиками [18]. Скоринг осуществляется с помощью «скоринговой модели» – т.е. своеобразных «весов», которые «взвешивают» математически выраженные характеристики объекта [19]. Автоматизация процесса скоринга может быть реализована с помощью специализированного программного обеспечения.

Скоринговая модель – это статистическая модель, оценивающая вероятность того, что наступит определенное событие (например, своевременное погашение долга заемщиком) [18]. В скоринговой модели используется набор факторов, характеризующих риск, связанный с наступлением события, и в результате получается пороговая оценка, которая и позволяет разделять события на «плохие» и «хорошие».

Для того чтобы лучше понять, что такое скоринг, далее приведено описание кредитного скоринга.

Кредитный скоринг – это процесс оценки заемщика банком или другой кредитной организацией. По результатам этой оценки потенциальный кредитор принимает решение по кредитной заявке. Если в ходе этого процесса, заемщик не набирает строго определенного количества баллов, то в получении кредита ему отказывают (табл. 4.1) [20].

Таблица 4.1

Управление выдачами кредита с использованием скоринговых баллов

|  |  |
| --- | --- |
| **Условие** | **Решение** |
| Скоринговый балл клиента | Отклонить |
| Скоринговый балл клиента | Одобрить |

Оцениваемые характеристики клиентов в случае кредитного скоринга могут быть следующие: пол, возраст, семейное положение, социальный статус (студент, пенсионер), стаж и другие. В табл. 4.2 представлен пример простой скоринговой карты.

Таблица 4.2

Простая скоринговая карта

|  |  |  |
| --- | --- | --- |
| **Характеристика** | **Атрибут** | **Балл** |
| Пол | М | 5 |
| Ж | 9 |
| Возраст |  | 2 |
|  | 12 |
| Семейное положение | В браке | 10 |
| Другое | 2 |

Этапы разработки скоринговой модели представлены на рис. 4.1.

Для сбора данных могут использоваться учетные системы, открытые источники информации, ввод данных вручную, проводиться маркетинговые исследования. Почти всегда производится предварительная обработка данных: определяются типы данных, удаляются ненужные данные, исправляются ошибки в данных и т.д. Когда данные получены, необходимо описать все факторы, потенциально влияющие на анализируемый процесс. Опрос экспертов, непосредственно владеющих проблемной ситуацией, поможет собрать максимум возможных предположений. На основании имеющихся данных согласно отобранным факторам формируются входные признаки (переменные), подаваемые на вход модели, и выходной (целевой) показатель, значение которого формируется на выходе модели как отклик на подачу на ее вход набора входных признаков.

К данным предъявляются следующие требования [21]:

* Количество примеров должно быть значительно больше количества факторов.
* Желательно, чтобы данные покрывали как можно больше ситуаций реального процесса.
* Исходные данные должны обладать определенной исторической давностью [17]. Какие данные считать актуальными, зависит от решаемой задачи.

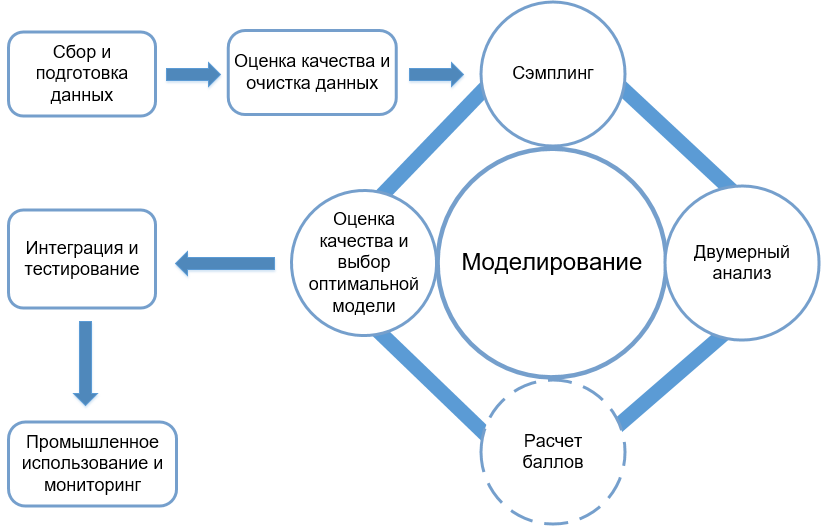


Рис. 4.1. Этапы разработки скоринговой модели

Далее проводится оценка качества данных и, если необходимо, очистка данных. В контексте современных аналитических технологий качество данных – совокупность их свойств и характеристик, определяющих степень пригодности для анализа [21]. Аналитические решения, полученные на основе некачественных данных, могут исказить реальную картинку исследуемых бизнес-процессов, показать ложные закономерности и связи объектов [22].

Типичный набор инструментов аналитического приложения позволяет обрабатывать: пропущенные значения (вызывают неопределенность при работе многих алгоритмов), аномалии (мешают обнаружению закономерностей в исследуемом процессе) – выбросы (значения данных, существенно отклоняющиеся от средних) и экстремальные значения (настолько сильно отклоняются от типичных значений, что перестают соответствовать логике исследуемых процессов), дубликаты (увеличивают объем выборки, но не повышают информативность данных), противоречия (одинаковым исходным данным соответствуют различные результаты, т.е. нарушаются общие закономерности в данных) [21].

Этапы сбора и очистки данных, как правило, являются длительными по времени и требуют привлечения большого количества ресурсов.

Сэмплинг (*Sampling*) – отбор из исходного множества данных (генеральной совокупности) с помощью специальных методов некоторого репрезентативного подмножества, например, для сокращения анализируемого объема данных или для получения обучающей и тестовой выборок. Репрезентативность – соответствие характеристик выборки характеристикам генеральной совокупности. Репрезентативность определяет, насколько возможно обобщать результаты исследования с привлечением определенной выборки на всю генеральную совокупность, из которой она была собрана [23]. Также специальные стратегии сэмплинга часто применяются для решения проблемы редкого класса, когда в исходных данных присутствует несбалансированность записей, соответствующих событиям и не событиям.

Двумерный анализ (*Bivariate analysis*) позволяет одновременно исследовать взаимоотношение двух переменных (входной и выходной) и в той или иной форме проверить гипотезы о причинных связах между ними [24]. Данная операция тесно связана с сокращением размерности данных, а именно с уменьшением числа разнообразных значений признаков за счет их объединения в пределах некоторого интервала (конечного класса) и присвоения интервалу определенной метки (рис. 4.2). Благодаря этому во многих случаях удается добиться существенного уменьшения вычислительных и временных затрат на аналитическую обработку данных с малым ущербом для информативности данных [21]. Этот процесс называется квантованием (*Binning*).

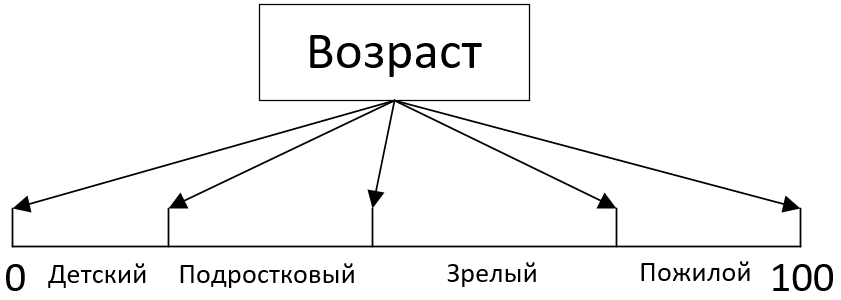


Рис. 4.2. Квантование значений признака

В процессе двумерного анализа может применяться метод *WoE*-анализа (*Weight of Evidence*), используемый для формирования конечных классов, где каждому наблюдению, содержащему набор признаков, ставится в соответствие бинарная выходная переменная [25]. Для каждого конечного класса признака вычисляется коэффициент *WoE*:

где *i* – индекс класса, – относительная частота появления не событий в классе, – относительная частота появления событий в классе. указывает на большую вероятность появления события, – не события.

На основе коэффициента *WoE* вычисляется информационный индекс *IV* (*Information Value*), определяющий значимость признака [26]:

Методика определения значимости [26]:

* – отсутствует;
* – низкая;
* – средняя;
* – высокая.

Коэффициенты *WoE* и *IV* – критерии для формирования конечных классов оптимальным способом [26]:

* максимизируя значимость признака в бинарной классификационной модели;
* максимизируя равномерность заполнения интервалов, что обеспечивает наилучшую репрезентативность результатов;
* компромисс между этими вариантами.

Когда получены обучающая и тестовая выборки, отобраны наиболее значимые факторы, производится обучение модели – процесс подбора (расчета) параметров модели (коэффициенты, веса нейронов, ветви дерева и т.д.) [24]. При необходимости, рассчитываются скоринговые баллы.

На этапе оценки качества полученные модели тестируются на контрольной выборке, рассчитываются различные метрики качества, строятся специализированные диаграммы. В результате выбирается оптимальная модель, которую затем можно интегрировать, например, в систему принятия решений. Далее осуществляется тестовая эксплуатация разработанной аналитической системы, по итогу которой принимается решение о промышленном использовании, и мониторинг (аудит модели на предмет ее актуальности).

Можно отметить следующие преимущества скоринговых систем:

* оценка результата события осуществляется практически мгновенно, как правило, на основе большого количества статистических данных;
* решение принимается непредвзято (независимо от настроения, опыта экспертов), опираясь только на математические правила;
* легкость интерпретации скоринговых баллов;
* прогнозный потенциал скоринговых моделей [27].

Недостатки:

* формирование выводов на основе статистических данных прошлых событий [27];
* скоринговые системы необходимо постоянно дорабатывать и поддерживать, поскольку они учитывают только прошлый опыт и реагируют на изменение обстоятельств с запозданием [28].

## 4.3. Описание модели логистической регрессии

В машинном обучении задача прогнозирования относится к задаче регрессии. Регрессия – установление зависимости непрерывной выходной переменной от входных переменных (рис. 4.3) [21].

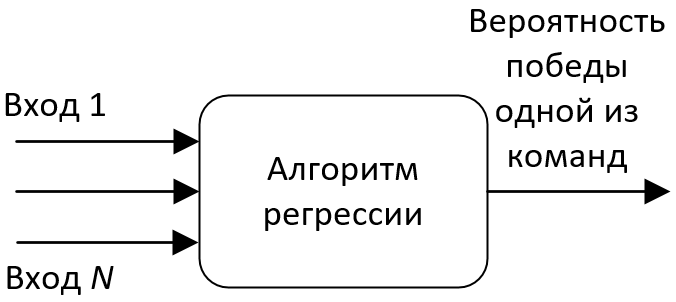


Рис. 4.3. Иллюстрация задачи регрессии

Логистическая регрессия – самая распространенная статистическая модель для построения скоринговых карт при бинарной зависимой переменной (т.е. может принимать только два значения). Математическая модель логистической регрессии выражает зависимость логарифма шанса от линейной комбинации независимых переменных [17]:

где *pi* – вероятность наступления события для *i*-го наблюдения, *xi(j)* – значение *j*-ой независимой переменной, *bj* – коэффициенты регрессии или параметры модели (*b0* – независимая константа), *y* – выходная (зависимая) переменная модели, – компонент случайной ошибки.

Шанс – вероятность наступления события, разделенная на вероятность наступления не события. Например, если вероятность победы первой команды в матче , то шансы на победу равны: . Тогда шансы на поражение первой команды равны: .

Отношение шансов – отношение шансов того, что событие произойдет, к шансам того, что событие не произойдет. В примере выше отношение шансов того, что победит первая команда, к тому, что она проиграет, равно: .

Для вычисления вероятности наступления события используется следующее регрессионное уравнение [29]:

где *p* – вероятность того, что произойдет интересующее событие, *e* – основание натурального логарифма, *y* – уравнение регрессии.

На рис. 4.4 показана зависимость, связывающая вероятность события и величину *y*. Эта функция определена на бесконечности и изменяется в диапазоне от 0 до 1.

Обучение модели подразумевает расчет коэффициентов *bj* регрессии. Значения коэффициентов отражают степень влияния независимых переменных на шанс появления события в логарифмической шкале. Коэффициенты логистической регрессии оцениваются на основе метода максимального правдоподобия, который позволяет найти такие значения коэффициентов, для которых вероятность появления максимальна [21]. Значения коэффициентов регрессии используются для расчета скоринговых баллов.

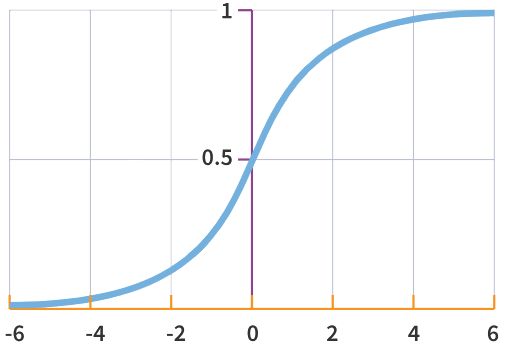


Рис. 4.4. Кривая логистической регрессии

## 4.4. Определение независимых и зависимых признаков

Первое, о чем нужно подумать при создании прогнозирующей модели, это то, на данных какого типа она будет строиться. Для того чтобы определить входные (независимые) переменные модели, необходимо ответить на следующий вопрос: «Какие факторы могут повлиять на вероятность победы определенной команды в матче?». На основе своего пятилетнего игрового опыта в *Dota 2* автором были выдвинуты следующие гипотезы:

* Винрейт (доля побед) команды за определенный период времени (за все время, за 12 месяцев, за 3 месяца) (*WINRATE, WINRATE12, WINRATE3*).
* Винрейт команды против команды соперников за 12 месяцев (*PWINRATE*).
* Среднее число смертей героев команды за матч за 12 месяцев (*DEATH\_AVG*).
* Доля матчей, в которых число смертей меньше или равно 25, за 12 месяцев (*DEATH\_L25*). Следует указать, что после сбора данных было вычислено, что среднее значение между медианой и средним числом смертей и убийств по всем имеющимся матчам равно 25.
* Среднее число убийств героев за матч за 12 месяцев (*KILL\_AVG*).
* Средняя продолжительность матчей в секундах, которые команда проиграла, за 12 месяцев (*LM\_AVG*).
* Средняя продолжительность матчей в секундах, которые команда выиграла, за 12 месяцев (*WM\_AVG*).
* Доля матчей, в которых команда проиграла и которые длились больше 35 минут, за 12 месяцев (*LM\_G35*). Следует указать, что после сбора данных было вычислено, что среднее между медианой и средней продолжительностью матча по всем имеющимся матчам равно 35 минутам.
* Доля матчей, в которых команда выиграла и которые длились больше 35 минут, за 12 месяцев (*WM\_G35*).
* Доля матчей, в которых команда выиграла и которые длились меньше 20 минут, за 12 месяцев (*WM\_L20*).
* Длительность существования команды в днях (*FOUNDED*).
* Количество сыгранных матчей командой за все время (*NMATCHES*).
* Суммарное количество баллов за игроков в команде согласно общему одиночному рейтингу (*TOP500*).
* Количество очков команды за турниры сезона *Dota Pro Circuit* (*DPC*).
* Средний винрейт за героев, выбранных командой в текущем матче, за 12 месяцев (*HWINRATE*).
* Средний винрейт против героев, выбранных командой соперников в текущем матче, за 12 месяцев (*HHWINRATE*).
* Общий показатель на основе двух предыдущих, рассчитывающийся как среднее значение (*HWR\_AVG*).
* Средняя эффективность героев, выбранных командой, против героев соперников согласно общедоступной статистике за 12 месяцев (*COUNTERS*).

Необходимо отметить, что выше указаны только те признаки, данные по которым имеются в открытом доступе и их можно загрузить из внешних источников.

Для примера, если команда за все время сыграла 73 матча и выиграла из них 49, то . *TOP500* – за игроков, входящих в первые пять сотен согласно общему одиночному рейтингу начисляется 3 балла, в следующие пять сотен – 2 балла, за остальных – 1 балл, за не попавших в рейтинг – 0 баллов.

Более подробно следует описать признак *COUNTERS*. На веб-ресурсе *Dotabuff.com* имеется статистика по каждому герою, насколько он эффективен против других героев, собранная на основе данных из миллионов матчей [6]. Эффективность героев друг против друга описывается коэффициентом, который рассчитывается определенным образом (точная формула на сайте не приводится). Значения близкие к нулю говорят о равноценности героев. Чем меньше значение, тем более эффективен один герой против другого и наоборот.

Пример логических рассуждений при выборе факторов: «Например, винрейт команды «A» за все время (т.е. в среднем победы в каждых 7,5 матчах из 10), а у команды «Б» – (победы в 3,7 матчах из 10), то, вероятнее всего, победит команда «А»». Другой пример: «Если в личных встречах команда «А» намного чаще побеждала команду «Б», то можно предположить, что команда «А» – неудобный соперник для команды «Б» и что в следующем матче победа за командой «А»». Третий пример: «Чем меньше в среднем смертей у команды за матч, то, скорее всего, команда опытная, умеет контролировать ситуацию по ходу игры, играет осторожно, а значит шансы выиграть увеличиваются. За каждую смерть герои соперников получают дополнительное золото и опыт, это может перевернуть ход игры». Четвертый пример: «Если согласно статистике большинство героев команды «А» слабее героев команды «Б», то с самого начала игры определенное преимущество имеет вторая команда. Профессиональные игроки, взглянув на выбранных героев перед началом матча, могут с большой точностью определить, у какой команды шансы на победу выше». Пятый пример: «Если в одной из команд явный перевес игроков, занимающих более высокие места в общем одиночном рейтинге, то, наверное, у нее шансы на победу выше, так как игроки более матерые».

Так как в матче участвуют две команды, то значение каждого признака необходимо рассчитывать для обеих команд. Тогда на вход модели должно подаваться значение, отражающее взаимосвязь между одним и тем же показателем каждой команды. В половине случаев используется разность двух значений. Например, если значения признака *TOP500* для команд, участвующих в матче, равны 10 и 4, то реальное значение этого фактора, подаваемое на вход модели, может быть равно и наоборот. Исключением является переменная *COUNTERS*, которая уже содержит в себе информацию о взаимосвязи между героями обеих команд, *PWINRATE*, которая учитывает взаимный винрейт команд, и *HWR\_AVG*, рассчитывающаяся на основе двух других переменных.

Для разрабатываемой модели событием будет являться победа первой команды в матче, соответственно не событие – поражение. Тогда зависимая переменная *event*, характеризующая наступление события или не события, будет принимать значение “*win*” или “*lose*”.

Следует сделать важное поясняющее замечание. В каждом матче происходит одновременно событие и не событие, поскольку одна команда выигрывает, а другая проигрывает. Этот факт в дальнейшем облегчит построение модели.

## 4.5. Сбор и подготовка данных

В феврале 2019 года с помощью скриптов, написанных на языке программирования *Python*, с веб-ресурса *Dotabuff.com* [6] были получены необходимые данные о командах, об игроках, об эффективности героев друг против друга. Важно было не упустить преимущественно те команды, которые существуют несколько лет, активны на текущий момент и сыграли как можно больше матчей. Поэтому выбирались команды, участвовавшие в крупных турнирах за 2017 и 2018 года. При этом отбрасывались команды, у которых последняя активность была до 9 марта 2017 года, когда состоялся первый большой турнир в этом году.

Интуитивно понятно, что чем больше данных используется для обучения, тем успешнее оно будет. Используя предоставляемый бесплатно веб-ресурсом *Opendota.com* *API* [7], были загружены в формате *JSON* данные о матчах команд от 1 мая 2016 года до 6 февраля 2019 года. Общий размер *JSON*-файлов на диске составил 6 Гб.

После загрузки данных необходимо было отсеять некорректные матчи, чтобы данные были максимально достоверными. Были написаны скрипты со следующими фильтрами:

1. Режим игры должен быть *Captain’s Mode*, который используется на турнирах.
2. Поиск матчей с продолжительностью не более 15 мин.
3. Поиск матчей с общим числом смертей/убийств героев менее 20.
4. Поиск матчей с разницей в счете больше чем 15.
5. Поиск матчей с разницей *GPM* (среднее количество золота в минуту, добываемого командой в матче) больше чем 1500.
6. Поиск матчей с крайними значениями *GPM*, длительности, числом смертей/убийств.

После анализа найденных потенциально некорректных матчей 518 из них были удалены.

Посчитав количество сыгранных матчей для каждой команды, можно отсеять некоторые команды и обнаружить новые, у которых сыграно не меньше определенного количества матчей. Для этого был написан скрипт, а нижний порог количества матчей для команды установлен равным 20. Для новых команд недостающие матчи были дозагружены, а матчи между неизвестными командами удалены. В итоге остались 277 команд и 29122 матча. Количество матчей только между этими командами составило 15567, что и обеспечило объем генеральной совокупности равный 31134 записям для обучения и тестирования модели.

Каждая запись представляет собой комбинацию «вход – выход», где «вход» – набор входных признаков, а «выход» – выходной признак (целевой). Пример двух записей с 5 входными переменными, описывающих один матч, представлен в табл. 4.3, где *ID* – идентификатор матча (информационная переменная, не участвующая в расчетах), *event­* – выходная переменная (событие или не событие). Первая запись соответствует победе одной из команд в матче, вторая – поражению.

Все записи (прецеденты) были сохранены в текстовый файл *“SAMPLE.txt”*, в котором каждый прецедент находится в новой строке, первая строка содержит названия столбцов, а значения в строках разделены символом табуляции (приложение А).

Признаки *COUNTERS* и *HWR\_AVG* имеют вещественный тип данных, *event* – строковый, все остальные – целый

Таблица 4.3

Пример записей для обучения модели

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***ID*** | ***WINRATE*** | ***FOUNDED*** | ***TOP500*** | ***DPC*** | ***DEATH\_AVG*** | ***event*** |
| 2332626658 | 6 | -130 | 10 | 90 | -4 | *win* |
| 4312625751 | -13 | 545 | -2 | 150 | 8 | *lose* |

## 4.6. Оценка качества и очистка данных

Для начала необходимо импортировать файл “*SAMPLE.txt*”, полученный на этапе сбора и подготовки данных, в программу *Deductor Studio*. Для этого используется мастер импорта, помогающий в интерактивном пошаговом режиме настроить соответствующие параметры. На рис. 4.5 изображен один из шагов импорта, где можно выбрать файл, указать кодировку и другое.

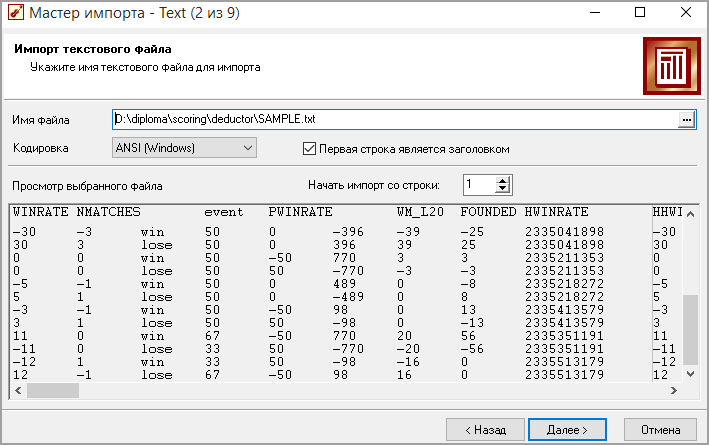


Рис. 4.5. Импорт данных в Deductor Studio

Для оценки качества данных в *Deductor Studio* используется обработчик (узел) «Качество данных». Он позволяет обнаружить пропуски в данных, выбросы и экстремальные значения.

На рис. 4.6 видно, что распределение значений данных близко к нормальному, поэтому в настройках обработчика в качестве метода определения выбросов и экстремальных значений было выбрано среднеквадратическое отклонение со значением «3» для выбросов и «5» для экстремальных. Таким образом, значение признака будет считаться аномальным, если оно отличается от среднего более, чем на заданное число стандартных отклонений.

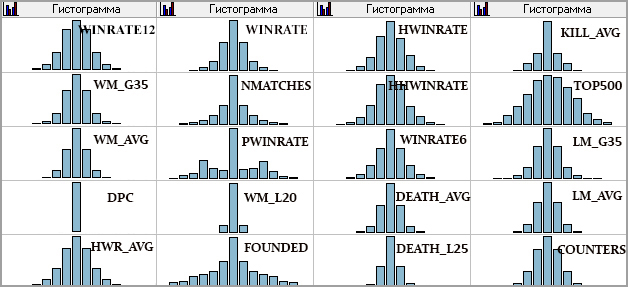


Рис. 4.6. Распределения значений признаков

Результат работы обработчика можно наблюдать на рис. 4.7. Пропуски в данных отсутствуют, у большинства факторов имеются выбросы и экстремальные значения, не требуют предобработки только признаки *FOUNDED* и *PWINRATE*.

Для редактирования аномальных значений используется обработчик «Редактирование выбросов и экстремальных значений», который получает информацию из обработчика «Оценка качества» и обрабатывает аномалии заданным способом. На рис. 4.7 видно, что в полях столбцов «Выбросы» и «Экстремальные» находится значение «Ограничивать», т.е. выявленные аномалии будут заменены пороговыми значениями выброса. Также можно выбрать значения «Оставить без изменения», «Удалять записи», «Заменять средним», «Заменять медианой» и «Заменять наиболее вероятным».

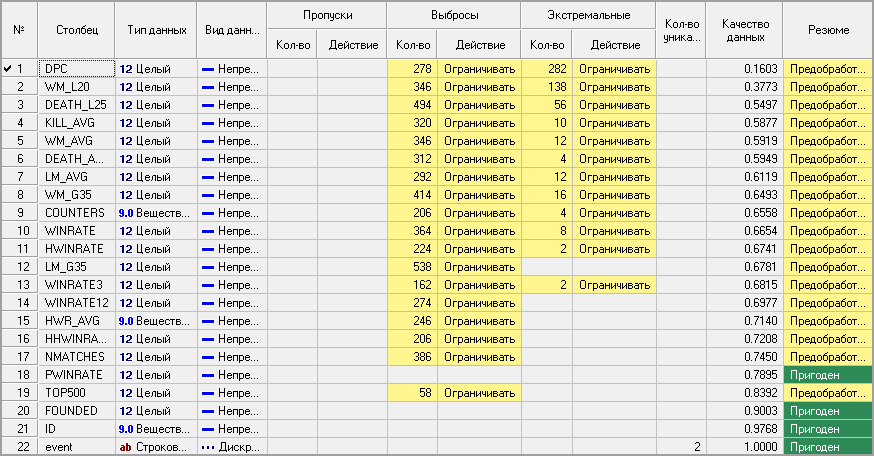


Рис. 4.7. Результат работы обработчика «Оценка качества»

Необходимо рассмотреть хотя бы один пример. Стандартное отклонение для параметра *WINRATE* равно 11,86, тогда все значения больше чем будут считаться аномальными. Например, среди таких значений имеется число 60, т.е. разница между винрейтами команд составила 60%. Такая ситуация маловероятна в реальности, поскольку винрейт команд, активно участвующих в турнирах, колеблется примерно в пределах от 35% до 70%. Все аномалии параметра *WINRATE* были заменены пороговыми значениями -33 и 33, соответственно для отрицательных и положительных значений, что ближе к реальности. Так были проанализированы аномальные значения для всех предикторов и принято решение о замене пороговыми значениями.

В табл. 4.4 представлены среднеквадратическое отклонение, минимум и максимум значений признаков.

Для поиска дубликатов и противоречий используется обработчик «Дубликаты и противоречия». Их обнаружено не было.

Таблица 4.4

Среднеквадратическое отклонение, минимум и максимум значений признаков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Признак** | **Мин** | **Макс** |  | **Признак** | **Мин** | **Макс** |  |
| *WINRATE3* | -43 | 43 | 14,59 | *WINRATE* | -33 | 33 | 11,54 |
| *WM\_G35* | -44 | 44 | 15,29 | *NMATCHES* | -568 | 568 | 198,13 |
| *WM\_AVG* | -620 | 620 | 215,7 | *PWINRATE* | 5 | 95 | 15,25 |
| *DPC* | -748 | 748 | 203,3 | *WML\_20* | -20 | 20 | 6,96 |
| *HWR\_AVG* | -39,49 | 39,49 | 13,57 | *FOUNDED* | -2327 | 2327 | 888,99 |
| *HWINRATE* | -45 | 45 | 15,39 | *KILL\_AVG* | -9 | 9 | 2,99 |
| *HHWINRATE* | -46 | 46 | 15,85 | *TOP500* | -14 | 14 | 4,66 |
| *WINRATE12* | -37 | 37 | 12,68 | *LM\_G35* | -46 | 46 | 16,22 |
| *DEATH\_AVG* | -10 | 10 | 3,33 | *LM\_AVG* | -639 | 639 | 220,99 |
| *DEATH\_L25* | -35 | 35 | 12,24 | *COUNTERS* | -0,81 | 0,81 | 0,28 |

## 4.7. Сэмплинг

Объем генеральной совокупности составил 31134 записей, что намного больше числа предикторов (их всего 20). При таком количестве прецедентов и предикторов обучение модели происходит всего за несколько секунд. Также отсутствует проблема редкого класса, поскольку распределение числа «хороших» и «плохих» наблюдений 50:50. По этим причинам нет необходимости сокращать объем исходной выборки на текущем шаге. Но следует отметить, что сэмплинг будет произведен далее для получения обучающей и тестовой выборок данных. Для применения сэмплинга в *Deductor Studio* используется обработчик «Сэмплинг».

## 4.8. Двумерный анализ признаков

Для дальнейшей работы необходимо добавить поле логического типа вместо переменной *event*, поскольку некоторые обработчики не поддерживают выходные поля строкового типа. Для этого используется узел «Калькулятор», позволяющий создавать новые поля, значения которых вычисляются по заданному выражению. По рис. 4.8 понятно, что поле *Result* будет иметь значение “*True*” для всех наблюдений, характеризующих события, и “*False*” – не события.

Обработчик «Конечные классы» позволяет производить квантование значений признаков на конечные классы и автоматически рассчитывает коэффициенты *WoE* и *IV* для каждого признака. В настройках обработчика указываются входные и выходные признаки, для каждого признака были заданы следующие настройки:

* максимальное количество классов – 5;
* минимальный вес класса (малый вес класса указывает на его низкую значимость и целесообразность объединения с некоторым другим классом) – 5%;
* равномерность (значения близкие к 0 обеспечивают разбиение на классы, которое максимизирует *IV*; значения близкие к 1 обеспечивают примерно одинаковое число наблюдений в каждом классе в ущерб *IV*) – 0;
* предквантование (задается начальное число интервалов, из которых будут сформированы конечные классы; рекомендуется использовать при большом числе уникальных значений) – 40.

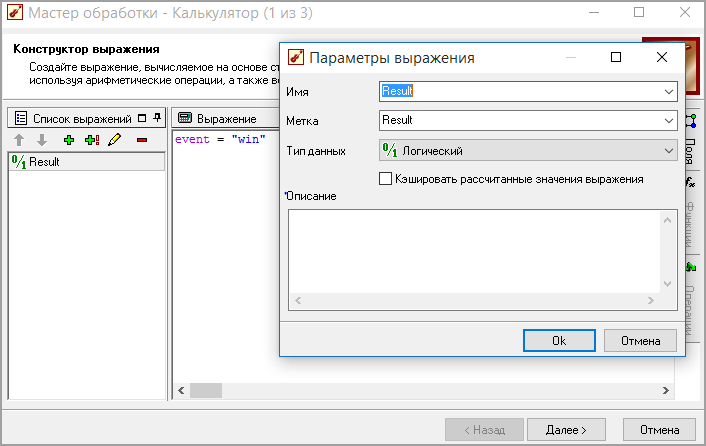


Рис. 4.8. Создание поля Result в обработчике «Калькулятор»

На рис. 4.9 и в табл. 4.5 представлены визуализация и таблица конечных классов для признака *WM\_AVG*. По вертикальной оси указаны значения коэффициента *WoE*, а по горизонтальной – номера начальных классов, красным цветом выделены конечные классы, сверху указаны их метки.

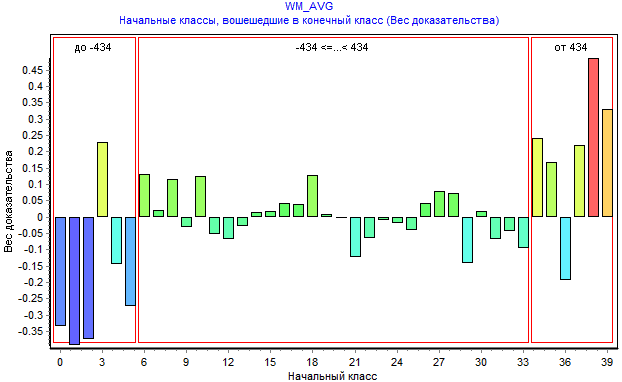


Рис. 4.9. Визуализация конечных классов для признака WM\_AVG

Таблица 4.5

Таблица конечных классов для признака WM\_AVG

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Конечные классы** | | **Целевая функция** | | | **Критерий** | |
| **Номер** | **Метка** | **События** | **Не события** | **Всего** | ***WoE*** | ***IV*** |
| 1 | до -434 | 502 | 397 | 899 | -0,234664 | 0,001583 |
| 2 | -434<=…<434 | 14660 | 14663 | 29323 | 0,000205 | 0,000000 |
| 3 | от 434 | 405 | 507 | 912 | 0,224624 | 0,001472 |
| **Всего** |  | 15567 | 15567 | 31134 |  | 0,003055 |

Расчет коэффициентов *WoE* и *IV*:

В табл. 4.5 видно, что первому конечному классу соответствует большее число событий, чем не событий, это можно интерпретировать следующим образом: «команды, которые в среднем выигрывают на 7 минут быстрее () соперников, немного чаще побеждают в матчах», обратная ситуация характерна для третьего интервала, но в эти интервалы попадает всего около 6% процентов всех наблюдений. Статистика по второму интервалу свидетельствует о равной вероятности появления событий или не событий, к тому же ему принадлежат оставшиеся записи, отсюда значения *WoE* и *IV* близкие к 0. В результате , поэтому значимость отсутствует. Если расширять первый и третий интервалы, которые более значимые, то разница между числом «хороших» и «плохих» записей получается меньше, чем есть сейчас, и вес доказательства уменьшается. Таким образом, можно сделать вывод, что данный фактор не следует использовать для построения модели.

После настройки конечных классов для всех предикторов были получены результаты, представленные на рис. 4.10.

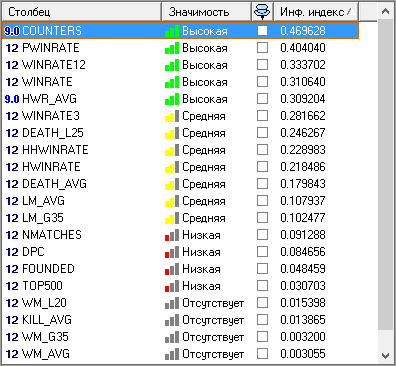


Рис. 4.10. Значимость предикторов

На данном шаге можно отсеять признаки с низкой информативностью: *NMATCHES*, *DPC*, *FOUNDED*, *TOP500*, *WM\_L20*, *KILL\_AVG*, *WM\_G35*, *WM\_AVG*.

## 4.9. Сокращение числа признаков

Признаки можно разделить на две группы [21]:

* Незначащие, которые слабо или никак не связаны с результатом анализа. Во-первых, они не несут в себе полезной информации. Во-вторых, незначащие данные могут содержать закономерности, не связанные с решаемой задачей, которые могут быть обнаружены моделью вместо искомых.
* Избыточные, которые содержат полезную информацию, но без которых могут быть получены хорошие результаты анализа. Т.е. избыточные признаки дублируют информацию в том или ином виде.

Незначащие признаки были обнаружены и исключены в процессе двумерного анализа. Чтобы выявить избыточные признаки, необходимо произвести дополнительный анализ.

Очевидно, что винрейт команды за все время включает в себя винрейт за 12 месяцев, 3 месяца и персональный винрейт против любой другой команды, а винрейт за 12 месяцев – винрейт за 3 месяца. Дополнительно, используя обработчик «Корреляционный анализ», можно проверить, имеется ли линейная функциональная зависимость между двумя входными параметрами.

Считается, что [21]:

* если коэффициент корреляции , то между *X* и *Y* существует функциональная линейная зависимость;
* если , то *X* и *Y* не коррелированы (но может иметь место нелинейная зависимость);
* если или , то связь между *X* и *Y* очень высока; если или , то зависимость отсутствует; промежуточные значения свидетельствуют о наличии определенной связи.

Согласно табл. 4.6 наименее коррелирован с другими параметрами признак *PWINRATE*, к тому же он имеет наибольшую значимость (рис. 4.10), поэтому его следует оставить. Также следует оставить предиктор *WINRATE12*, поскольку он имеет наибольшую значимость среди оставшихся и уже включает в себя всю необходимую информацию. Можно рассуждать следующим образом: «составы команд меняются, игра *Dota 2* ежемесячно обновляется, т.е. данные устаревают, поэтому не рекомендуется использовать винрейт за все время (два с половиной года), а за 3 месяца команда может сыграть очень мало матчей».

Таблица 4.6

Корреляционный анализ предикторов WINRATE, WINRATE12, WINRATE3, PWINRATE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Признак** | ***WINRATE*** | ***WINRATE12*** | ***WINRATE3*** | ***PWINRATE*** |
| ***WINRATE*** | - | 0,877 | 0,741 | 0,316 |
| ***WINRATE12*** | 0,877 | - | 0,834 | 0,298 |
| ***WINRATE3*** | 0,741 | 0,834 | - | 0,288 |
| ***PWINRATE*** | 0,316 | 0,298 | 0,288 | - |

Также взаимосвязаны между собой пары признаков *DEATH\_L25* и *DEATH\_AVG*, *LM\_G35* и *LM\_AVG* (табл. 4.7). Например, сильную обратную связь между *DEATH\_L25* и *DEATH\_AVG* можно объяснить так: «Чем больше матчей с количеством смертей , тем меньше среднее число смертей по всем матчам». Ориентируясь на коэффициент *IV* (рис. 4.10), логично оставить признаки *DEATH\_L25* и *LM\_AVG*.

Таблица 4.7

Корреляционный анализ предикторов DEATH\_L25, DEATH\_AVG, LM\_G35, LM\_AVG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Признак** | ***DEATH\_L25*** | ***DEATH\_AVG*** | ***LM\_G35*** | ***LM\_AVG*** |
| ***DEATH\_L25*** | - | -0,866 | 0,113 | 0,079 |
| ***DEATH\_AVG*** | -0,866 | - | -0,02 | -0,052 |
| ***LM\_G35*** | 0,113 | -0,02 | - | 0,806 |
| ***LM\_AVG*** | 0,079 | -0,052 | 0,806 | - |

Предиктор *HWR\_AVG* – это общий признак, содержащий информацию о *HWINRATE* и *HHWINRATE*. За счет этого объединения удалось повысить значимость до высокой (рис. 4.10) и понизить размерность данных, оставив только фактор *HWR\_AVG*. Действительно, ведь если команда имеет плохую статистику за выбранных героев в текущем матче, а команда соперников за своих героев – намного лучше, то можно предположить, что первой команде придется приложить больше усилий для победы, чем если бы у команды соперников статистика была такая же плохая.

В результате отбора наиболее значащих признаков остались следующие в порядке уменьшения значимости: *COUNTERS*, *PWINRATE*, *WINRATE12*, *HWR\_AVG*, *DEATH\_L25*, *LM\_AVG*. В табл. 4.8 выделяется сильная корреляция между признаками *WINRATE12* и *HWR\_AVG*. Данную ситуацию можно интерпретировать следующим образом: «Доля побед у команды тем больше, чем, скорее всего, чаще она выбирает своих сигнатурных героев, а команды соперников реже в тех же матчах». Также имеются определенные статистические связи между параметрами *WINRATE12* и *DEATH\_L25*, *HWR\_AVG* и *DEATH\_L25*, которые можно попытаться объяснить. В любом случае было решено оставить все шесть предикторов (при обучении модели опытным путем было установлено, что удаление любого признака ухудшает модель в той или иной степени).

Существуют более сложные методы сокращения размерности данных, например, метод главных компонент, но их рассмотрение выход за рамки данной работы.

Таблица 4.8

Корреляционный анализ предикторов

COUNTERS, PWINRATE, WINRATE12, HWR\_AVG, DEATH\_L25, LM\_AVG

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Признак** | ***COUNTERS*** | ***PWINRATE*** | ***WINRATE12*** | ***HWR\_AVG*** | ***DEATH\_L25*** | ***LM\_AVG*** |
| ***COUNTERS*** | - | -0,023 | -0,001 | 0,014 | -0,01 | 0,03 |
| ***PWINRATE*** | -0,023 | - | 0,298 | 0,288 | 0,185 | 0,18 |
| ***WINRATE12*** | -0,001 | 0,298 | - | 0,703 | 0,476 | 0,215 |
| ***HWR\_AVG*** | 0,014 | 0,288 | 0,703 | - | 0,408 | 0,203 |
| ***DEATH\_L25*** | -0,01 | 0,185 | 0,476 | 0,408 | - | 0,079 |
| ***LM\_AVG*** | 0,03 | 0,18 | 0,215 | 0,203 | 0,079 | - |

## 4.10. Разбиение на обучающее и тестовое множества

Обучающая и контрольная выборка должны формироваться на основе механизма случайного отбора обычно в соотношении 70-80% и 30-20% соответственно от исходного объема исторической выборки [17]. Известно, что выборка должна быть репрезентативной, притом настолько, насколько это возможно. Один из показателей репрезентативности, чтобы распределение на выборке было близко к тому распределению, которое есть на генеральной совокупности.

Обработчик «Разбиение на множества» позволяет разбить исходное множество данных правильным образом на обучающую и тестовые выборки с

помощью различных методов сэмплинга [30]:

* Случайный: выборка производится случайным образом из всей совокупности.
* Равномерный случайный: все записи исходной совокупности разделяются на группы с одинаковым числом записей. Затем из каждой группы осуществляется выборка записей случайным образом.
* Стратифицированный: сначала выполняется группировка элементов исходной совокупности в относительно однородные группы (страты, слои). Затем осуществляет случайная выборка из каждого слоя по отдельности.
* Последовательный: последовательный отбор записей из всей совокупности, пока не будет достигнут требуемый объем.

Значения признаков неравномерно распределены среди прецедентов генеральной совокупности, поэтому используется стратифицированный метод, обеспечивающий наличие в выборке всех важных подгрупп. В качестве полей, определяющих страты, были выбраны шесть предикторов, отобранных на предыдущем шаге, и выходная переменная *Result*. Но поскольку опыт показывает, что использовать нужно не больше шести переменных [31], были оставлены только *Result*, *WINRATE12*, *DEATH\_L25* и *LM\_AVG*, обеспечивающие наибольшую схожесть групп в обучающем множестве.

Количество записей в обучающем и тестовом множествах выбрано 80% и 20% соответственно от исходной совокупности. Для примера, на рис. 4.11 представлена гистограмма распределения значений признака *DEATH\_L25*, вошедших в обучающее множество. В каждой паре столбцов левый соответствует исходному множеству, а правый – обучающему.

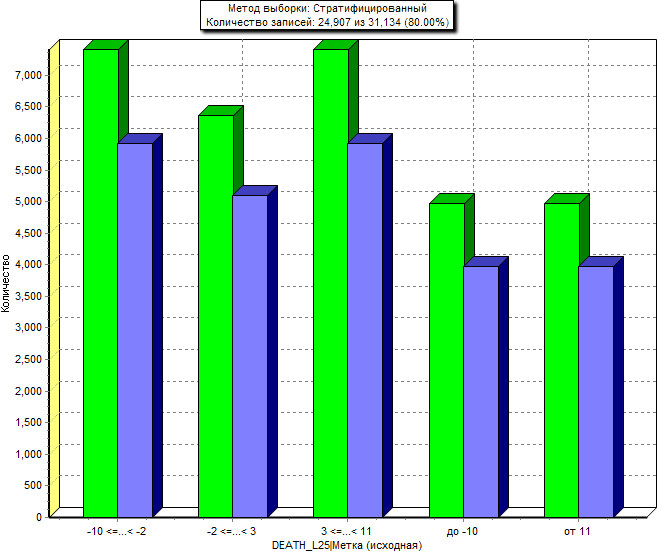


Рис. 4.11. Распределение значений признака DEATH\_L25 в обучающем множестве

В результате в обучающую выборку вошло 24907 записей (80%), в тестовую – 6227 (20%).

## 4.11. Оценка качества и интерпретация модели

Для обучения модели в *Deductor Studio* используется обработчик «Логистическая регрессия». В первую очередь в настройках выбираются входные и выходная переменные. В модель необходимо включить все переменные, которые позволяют повысить точность оценок, получаемых с помощью регрессии. Узел «Логистическая регрессия» предоставляет следующие методы отбора:

* Полное включение (в модель включаются все доступные переменные).
* Прямой отбор (на каждом шаге в модель включается по одной переменной, если при этом повышается значимость модели).
* Обратное исключение (из начальной модели, состоящей из всех переменных, на каждом шаге исключается одна переменная, если выполняется условие исключения переменной).
* Последовательный отбор – модификация прямого отбора (на каждом шаге после включения в модель новой переменной проверяется значимость ранее введенных переменных, обнаруженные не значимые переменные исключаются).

При обучении модели опытным путем было установлено, что удаление любого признака ухудшает модель в той или иной степени, поэтому используется метод полного включения.



### 4.11.1. ROC-анализ

*ROC*-анализ основан на использовании *ROC*-кривой (*Receiver Operator Characteristic*), отражающей зависимость количества верно классифицированных положительных примеров от количества неверно классифицированных отрицательных примеров [32].

При уменьшении значения параметра модели – точки отсечения (*Cut-off value*), увеличивается вероятность ошибочного распознавания положительных примеров (ложноположительных исходов), а при увеличении возрастает вероятность неправильного распознавания отрицательных примеров (ложноотрицательных исходов). Фактически, точка отсечения – это значение уравнения регрессии, изменяется от 0 до 1.

В табл. 4.9 представлены результаты обучения и тестирования модели на обучающем и контрольном множествах соответственно при пороговом значении равном 0,51. Это значит, что модель будет классифицировать примеры как положительные (победа первой команды), если рассчитанная вероятность наступления события больше 0,51, в противном случае – как отрицательные (поражение первой команды).

Таблица 4.9

Таблица сопряженности результатов классификации моделью

на обучающем и тестовом множествах

|  |  |  |  |
| --- | --- | --- | --- |
| **Фактически** | **Классифицировано** | | |
| ***False*** | ***True*** | **Итого** |
| ***False*** | 10983 (*TN*) | 4584 (*FP*) | 15567 |
| ***True*** | 4920 (*FN*) | 10647 (*TP*) | 15567 |
| **Итого** | 15903 | 15231 | 31134 |

В табл. 4.9:

* *TP* (*True Positives*) – верно классифицированные положительные примеры (истинно положительные);
* *TN* (*True Negatives*) – верно классифицированные отрицательные примеры (истинно отрицательные);
* *FN* (*False Negatives*) – положительные примеры, классифицированные как отрицательные (ложно отрицательные, ошибка I рода);
* *FP* (*False Positives*) – отрицательные примеры, классифицированные как положительные (ложно положительные, ошибка II рода).

Точность модели (*Accuracy*) можно оценить с помощью общего показателя успеха (*Overall Success Rate*) – числа правильно классифицированных наблюдений [21]:

Тогда общий показатель ошибки (*Overall Error Rate*) составит [21]:

В табл. 4.10 представлены результаты классификации моделью на обучающем множестве, а в табл. 4.11 – на тестовом. Согласно табл. 4.10 и 4.11 и или ошибки обучения и обобщения соответственно примерно одинаковы, что говорит об отсутствии проблемы переобучения, когда модель хорошо объясняет примеры из обучающей выборки, но относительно плохо – примеры, не участвовавшие в обучении (тестовое множество). Доля правильных распознаваний на обучающем и тестовом множествах почти 70%, поэтому можно считать, что модель обладает хорошей способностью к обобщению в рамках решаемой задачи.

Таблица 4.10

Таблица сопряженности результатов классификации моделью

на обучающем множестве

|  |  |  |  |
| --- | --- | --- | --- |
| **Фактически** | **Классифицировано** | | |
| ***False*** | ***True*** | **Итого** |
| ***False*** | 8799 (*TN*) | 3652 (*FP*) | 12451 |
| ***True*** | 3904 (*FN*) | 8552 (*TP*) | 12456 |
| **Итого** | 12703 | 12204 | 24907 |

Таблица 4.11

Таблица сопряженности результатов классификации моделью

на тестовом множестве

|  |  |  |  |
| --- | --- | --- | --- |
| **Фактически** | **Классифицировано** | | |
| ***False*** | ***True*** | **Итого** |
| ***False*** | 2184 (*TN*) | 932 (*FP*) | 3116 |
| ***True*** | 1016 (*FN*) | 2095 (*TP*) | 3111 |
| **Итого** | 3200 | 3027 | 6227 |

Для дальнейшей оценки эффективности модели понадобятся чувствительность модели (*Sensitivity*) или доля истинно положительных случаев [32]:

и специфичность (*Specificity*) – доля истинно отрицательных случаев [32]:

Модель с высокой чувствительностью чаще обнаруживает положительные примеры, а с высокой специфичностью – отрицательные. Например, если разрабатываемая модель используется для принятия решений в ставках на киберспортивные события (на победу одной из команд) и игрок старается не рисковать, то специфический классификатор будет выявлять такие положительные случаи, когда одна из команд победит почти наверняка согласно оценке, при этом повысится доля верно классифицированных отрицательных (игрок не теряет свои деньги) и ложно отрицательных случаев (упущенная выгода).

На рис. 4.13 представлены *ROC*-кривые для обучающего и тестового множеств, которые практически совпадают. По оси *Y* расположена чувствительность *Se*, а по оси *X* – (доля ложно положительных случаев).

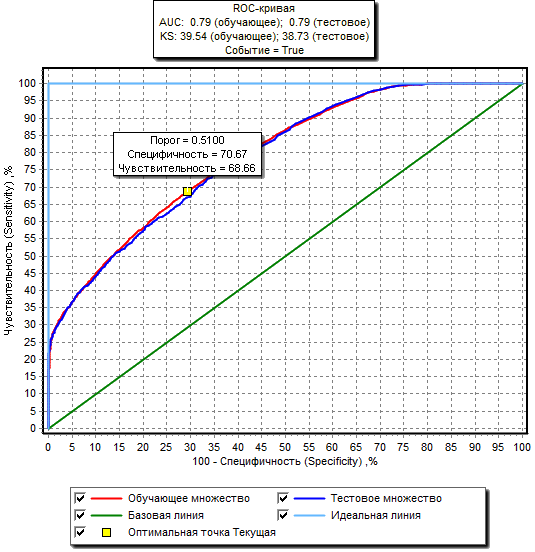


Рис. 4.13. ROC-кривые для обучающего и тестового множеств

Кривая, проходящая через верхний левый угол, соответствует идеальному классификатору, когда доля истинно положительных случаев составляет 100% (идеальная чувствительность), а доля ложно положительных равна 0 (идеальная специфичность). Диагональная линия соответствует случайному классификатору. Таким образом, чем больше под кривой площадь, тем больше прогностическая сила модели. Численный показатель площади называется *AUC* (*Area Under a Curve*), изменяется от 0,5 до 1.

В литературе иногда приводится следующая экспертная шкала значений *AUC* для оценки качества модели [32]:

* 0,9-1 – отличное;
* 0,8-0,9 – очень хорошее;
* 0,7-0,8 – хорошее;
* 0,6-0,7 – среднее;
* 0,5-0,6 – неудовлетворительное.

На рис. 4.13 видно, что , тогда согласно оценкам выше модель демонстрирует хорошую точность. Но следует отметить, что данный показатель скорее предназначен для сравнительного анализа нескольких моделей.

Невозможно одновременно повысить показатели *Se* и *Sp* модели, поэтому находят компромисс с помощью точки отсечения. Например, это может быть требование баланса (*Se Sp*) или максимизации суммы значений *Se* и *Sp*. На рис. 4.14 видно, что баланс достигается при пороговом значении равном 0,5001. Суммарный максимум достигается в точке отсечения равной 0,4911, тогда (т.е. 69,59% реальных побед одной из команд в матче будут подтверждены моделью), (т.е. 29,4% реальных поражений одной из команд в матче будут ошибочно классифицированы моделью как победы), а общая точность модели составит .

При анализе качества модели часто вычисляют индекс Джини (*Coefficient Gini*) по формуле [33]:

Существует устойчивое представление о том, какой диапазон индекса Джини соответствует удачным или неудачным моделям [34]:

* 0-0,2 – низкое качество модели;
* 0,2-0,4 – среднее;
* 0,4-0,6 – хорошее;
* 0,6-1 – отличное.

Поскольку , следовательно модель можно считать хорошей.

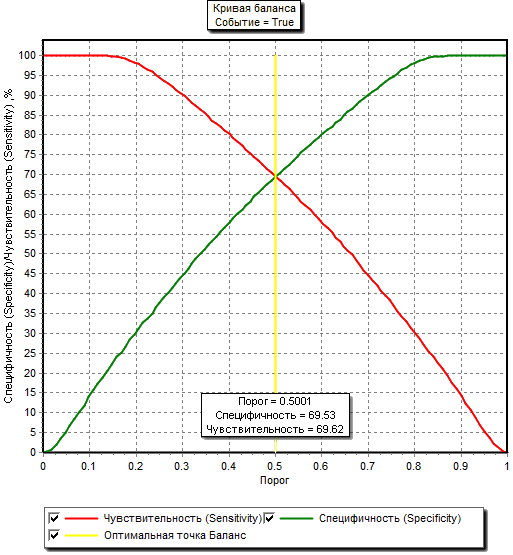


Рис. 4.14. Кривая баланса для обучающего и тестового множеств

### 4.11.2. Проверка значимости и интерпретация модели

В *Deductor Studio* для нахождения коэффициентов логистической регрессии используется метод максимального правдоподобия. Основу метода составляет функция правдоподобия (*Likelihood function*). На практике значение функции правдоподобия преобразуют через минус удвоенное значение логарифма правдоподобия [17]. Для проверки значимости логрегрессионной модели может использоваться тест отношения правдоподобия (*LR test*) [34]:

где *L1* – найденное значение логарифмической функции правдоподобия для модели со всеми входными переменными, *L0* – значение логарифмической функции правдоподобия для нулевой модели (с включением только константы). *LR* имеет распределение хи-квадрат с числом степеней свободы, равным числу независимых ограничений.

В табл. 4.12 представлен фрагмент отчета по регрессии из узла «Логистическая регрессия». Значение в столбце «Хи-квадрат» – это и есть тест отношения правдоподобия. Число степеней свободы равно 24, поскольку имеется 6 факторов, для каждого из которых 5 ограничений (табл. 4.13) и в каждой группе одному из них соответствует отношение шансов равное единице (опорная категория), отсюда . Сравнив рассчитанное значение с критическим значением из таблицы распределения хи-квадрат, получится , что позволяет отклонить нулевую гипотезу и сделать вывод о статистической значимости модели по сравнению с нулевой моделью.

Таблица 4.12

Фрагмент отчета по регрессии из обработчика «Логистическая регрессия»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **-*2Log Likelihood*** | **Хи-квадрат** | ***df*** | ***R2 McFadden*** | **Значимость** | **Метод отбора переменных** |
| 26625,134 | 7903,298 | 24 | 0,229 | 0,0000 | Полное включение |

Коэффициент *R2McFadden* – аналог коэффициента детерминации для линейной регрессии, который показывает, какую долю информации о поведении выходной переменной можно объяснить входными переменными, изменяется от 0 до 1 [21]. Чем ближе значение к 1, тем больше соответствие уравнения регрессии реальным данным. Тогда согласно этому показателю модель слабо соответствует реальным данным. Однако для модели логистической регрессии коэффициент детерминации не является основной характеристикой точности модели в отличие от модели линейной регрессии, и низкие значения коэффициента детерминации в модели логистической регрессии – это нормальной явление [17].

В табл. 4.13 представлен второй фрагмент отчета по регрессии с данными о коэффициентах и входных переменных. Для проверки значимости отдельной входной переменной применяется тест Вальда (*Wald test*) [21]:

где – коэффициент регрессии при данной входной переменной, – стандартная ошибка оценивания коэффициента регрессии на основе наблюдаемых данных. При нулевой гипотезе , данное отношение соответствует распределению хи-квадрат с одной степенью свободы.

Например, коэффициент Вальда для переменной *WINRATE12* с ограничением равен: . Тогда , т.е. вероятность справедливой нулевой гипотезы меньше 1%. Данное *p*-значение достаточно мало, поэтому переменная *WINRATE12* на данном интервале значима при определении вероятности победы одной из команд в матче. В табл. 4.13 видно, что все остальные значения из столбца «Значимость» меньше, чем 0,0069, а значит все входные переменные модели значимы.

Все 6 предикторов – это категориальные переменные, поэтому при расчете вероятности наступления события в уравнении регрессии вместо значений переменных будет стоять единица, помноженная на коэффициент для данной переменной.

95%-ый доверительный интервал означает, что истинное значение случайной величины с вероятностью 95% лежит в пределах данного интервала [21]. Отношение шансов (*OR*) для коэффициента *bi* рассчитывается следующим образом: . Если отношение шансов равно 1 (шансы наступления события и не события равны), то входная переменная при данном коэффициенте не значима, поскольку . В табл. 4.13 видно, что ни один доверительный интервал отношения шансов не содержит , значит с достоверностью 95% отношения шансов статистически значимы. Отрицательные коэффициенты регрессии дают , т.е. шансы наступления не события больше, чем шансы благоприятного исхода, и наоборот для положительных коэффициентов, когда . Таким образом, чем сильнее *OR* отличается от 1, тем более или менее значимой будет входная переменная (а значит и модель в целом).

Таблица 4.13

Фрагмент отчета по регрессии из обработчика «Логистическая регрессия»



Согласно табл. 4.13 значение отношения шансов для переменной «*PWINRATE* | 55 <=...< 65» составляет . Это значит, если винрейт первой команды против второй команды принимает значение из интервала , то шансы победить у первой команды при фиксированных значениях других переменных в 1,51303 раза выше, чем если бы винрейт принимал значение из интервала . То есть шансы на победу оцениваются относительно опорной категории, когда вероятность победы равна 0,5, поскольку . А если винрейт 65% или больше, то шансы увеличиваются в 3,28189 раза. Отношение шансов для «*COUNTERS* | от 0,4059» , значит шансы победить для команды, которая согласно статистике выбрала сильно неудачную комбинацию героев относительно героев соперников, примерно в 8 раз меньше, чем если бы соотношение эффективности между героями было примерно одинаковым – из интервала .

Общий вид получившегося уравнения логистической регрессии:

Пусть необходимо оценить вероятность победы команды «*EG*» в матче против команды «*TS*». Имеются следующие данные:

* Команда «*EG*» в личных встречах против команды «*TS*» одержала 7 побед в 10 матчах. Винрейт за последние 12 месяцев составляет 49%. Доля матчей, в которых число смертей меньше или равно 25, – 23%. Средняя продолжительность матчей, которые команда проиграла, – 35 минут или 2100 секунд. Средняя успешность героев, выбранных командой, против героев команды «*TS*» в текущем матче: -0,075. Значение признака, связывающего текущий винрейт двух команд за выбранных героев и против героев соперников, равен 14. Таким образом: , , , , , .
* Статистика команды «*TS*»: , , .
* Окончательные значения факторов для оценки вероятности победы команды «*EG*»: , , , , , .

Тогда вероятность победы команды «*EG*» составит:

За счет побед в личных встречах, факторов *HWR\_AVG* и *LM\_AVG* шансы на победу у команды «*EG*» оказались выше. Коэффициент *bi* при *COUNTERS* равен 0, т.е. шансы выиграть у обеих команд равны согласно данному признаку.

### 4.11.3. Оценка качества скоринговой карты

Для автоматического перевода коэффициентов логистической регрессии в скоринговые баллы в обработчике «Логистическая регрессия» необходимо задать следующие параметры масштабирования:

* *PDO* – количество баллов, которое удваивает шансы наступления (ненаступления) события;
* *Score* – количество баллов, в котором достигается отношение шансов (*Odds*) не событий к событиям.

Выбраны были следующие значения параметров, которые часто встречаются на практике [34]: *Score* = 600, *Odds* = 50:1, *PDO* = 20. В процессе вычислений рассчитываются множитель *f*:

и смещение:

Коэффициенты переводятся в баллы следующим образом, например, для *b0*:

В табл. 4.14 представлена полученная скоринговая карта. У константы указано количество соответствующих ей баллов плюс смещение.

Таблица 4.14

Скоринговая карта

|  |  |  |  |
| --- | --- | --- | --- |
| **Фактор** | **Балл** | **Фактор** | **Балл** |
| *PWINRATE* | 35 <=...< 45 | 0 | *DEATH\_L25* | -10 <=...< -2 | 0 |
| *PWINRATE* | 45 <=...< 55 | 5,8521 | *DEATH\_L25* | -2 <=...< 3 | 7,3101 |
| *PWINRATE* | 55 <=...< 65 | 11,9487 | *DEATH\_L25* | 3 <=...< 11 | 13,581 |
| *PWINRATE* | до 35 | -22,8237 | *DEATH\_L25* | до -10 | -9,5648 |
| *PWINRATE* | от 65 | 34,2906 | *DEATH\_L25* | от 11 | 23,4778 |
| *WINRATE12* | -12 <=...< -3 | 0 | *LM\_AVG* | -174 <=...< -50 | 0 |
| *WINRATE12* | -3 <=...< 4 | 3,3442 | *LM\_AVG* | -50 <=...< 43 | 3,9331 |
| *WINRATE12* | 4 <=...< 13 | 6,8251 | *LM\_AVG* | 43 <=...< 167 | 6,1542 |
| *WINRATE12* | до -12 | -10,8015 | *LM\_AVG* | до -174 | -4,8151 |
| *WINRATE12* | от 13 | 18,0986 | *LM\_AVG* | от 167 | 11,3771 |
| *COUNTERS* | -0,08118 <=...< 0,08118 | 0 | *HWR\_AVG* | -17,7733 <=...< -3,9496 | 0 |
| *COUNTERS* | -0,4059 <=...< -0,08118 | 20,9127 | *HWR\_AVG* | -3,9496 <=...< 3,9496 | 4,3093 |
| *COUNTERS* | 0,08118 <=...< 0,4059 | -22,017 | *HWR\_AVG* | 3,9496 <=...< 17,7733 | 7,584 |
| *COUNTERS* | до -0,4059 | 58,1086 | *HWR\_AVG* | до -17,7733 | -11,4541 |
| *COUNTERS* | от 0,4059 | -59,2408 | *HWR\_AVG* | от 17,7732 | 19,6106 |
| Константа | 463,7657 |  |  |

Для примера, который приведен выше, пусть необходимо определить, победит ли команды «*EG*» в матче против команды «*TS*». Тогда общий скоринговый балл составит:

Если значение порога отсечения установлено, например, в 485 баллов, значит команда «*EG*» победит согласно оценке, поскольку .

На рис. 4.15 представлена гистограмма распределения итоговых скоринговых баллов для всех записей. Нетрудно заметить, распределение имеет холмообразную форму, напоминающую кривую нормального распределения, а значит параметры масштабирования подобраны удачно [34].

На рис. 4.16 можно наблюдать распределение скоринговых баллов для «плохих» и «хороших» наблюдений по отдельности. В отличие от «хороших» записей распределение значений для «плохих» записей сконцентрировано в области низких скоринговых баллов. Идеальной считается скоринговая карта, при которой распределения баллов для событий и не событий не пересекаются, а находятся рядом друг с другом [34]. Видно, что площадь области пересечения графиков примерно равна площади двух смежных областей, тогда можно считать скоринговую карту приемлемой в рамках решаемой задачи.

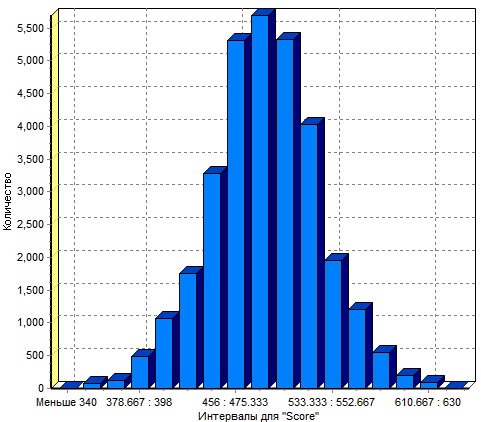


Рис. 4.15. Распределение скоринговых баллов

Для оценки качества прогнозирования скоринговой модели может использоваться статистика Колмогорова-Смирнова [33]. Вычисляется как максимальная разница между кумулятивными функциями распределения «плохих» и «хороших» наблюдений:

где *F(x)* – процент «плохих» записей со скорбаллом не выше *x*, *G(x)* – процент «хороших» записей со скорбаллом не выше *x*. На рис. 4.17 видно, что , что соответствует среднему качеству скоринговой карты согласно следующей шкале [34]:

* 0-30 – низкое качество;
* 30-40 – среднее;
* 40-50 – хорошее;
* 50-100 – отличное.

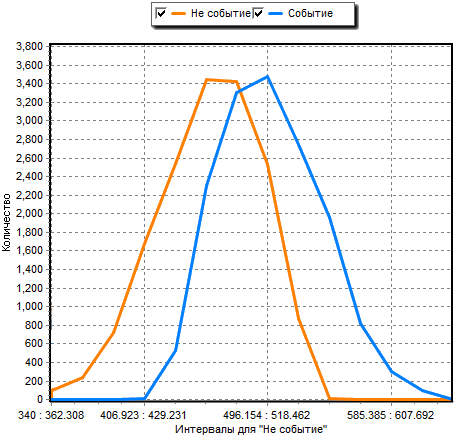


Рис. 4.16. Распределения скоринговых баллов для «хороших» и «плохих» наблюдений

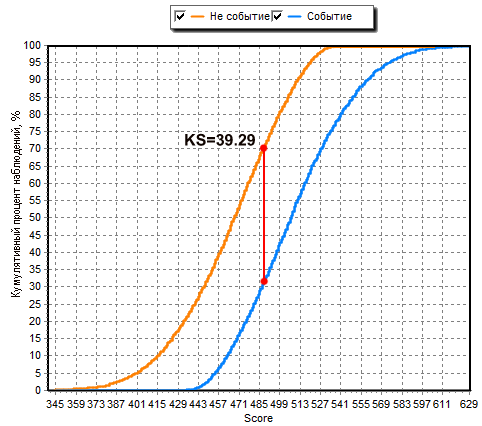


Рис. 4.17. Кумулятивные распределения скоринговых баллов для «хороших»

и «плохих» наблюдений

Также автоматически рассчитанные значения статистики Колмогорова-Смирнова и при построении *ROC*-кривых для обучающего и тестового множеств свидетельствуют о среднем качестве разработанной скоринговой модели (рис. 4.13).

## 4.12. Результаты и рекомендации по улучшению модели

В результате проделанной работы разработана скоринговая модель для оценки вероятности выигрыша одной из команд в игре *Dota 2* с помощью программы *Deductor Studio*. Общая точность модели составила 69,47%, а значит согласно техническому заданию разработанную модель можно считать успешной. Полученный результат не хуже некоторых моделей, упомянутых в сравнительном анализе.

Процесс построения скоринговой модели на основе математико-статистического метода – логистической регрессии изучен с теоретической и практической стороны. Пройдены этапы сбора и подготовки данных, оценки их качества и очистки, на основе двумерного анализа отобраны наиболее значимые признаки, сформированы обучающее множество, на котором модель обучалась, и контрольное, на котором затем была протестирована, построена скоринговая карта и в конце произведен анализ качества разработанной скоринговой модели. На рис. 4.18 изображен созданный в программе *Deductor Studio* скрипт построения модели.

Согласно оценке прогностической силы модели на основе *ROC*-анализа, проверке значимости модели с помощью различных критериев, оценке качества скоринговой карты можно сделать вывод о приемлемом качестве модели в целом. Было установлено, что при выборе значения порога отсечения, равного 0,4911, максимальная общая точность модели составляет 69,6% или 21669 правильно классифицированных из 31134 записей. Возникает вопрос, какие причины повлияли на такой результат и можно ли улучшить прогностическую силу модели.

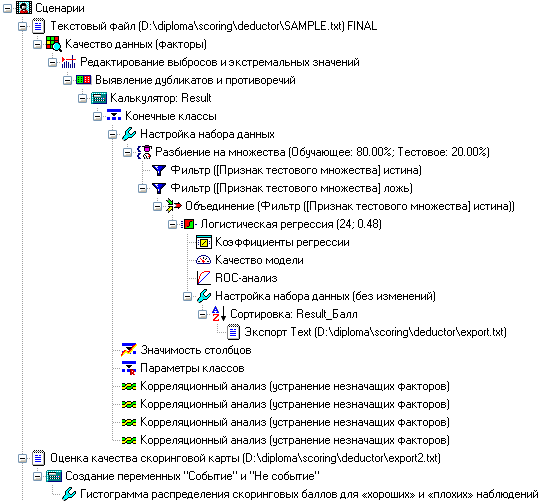


Рис. 4.18. Скрипт построения модели в программе Deductor Studio

Во-первых, следует отметить, что некоторые потенциально значащие факторы не рассматривались. Например, это может быть признак, учитывающий стратегию команд на основе выбранных ими героев. Всего героев 117, а значит существует более 167 миллионов комбинаций из 5 различных героев. Необходимо проводить анализ на предмет того, как различные герои сочетаются друг с другом.

Во-вторых, используется всего 6 категориальных признаков, некоторые из которых, как было установлено, в определенной степени коррелированы между собой. В реальных моделях, применяемых на практике, используется намного больше переменных, а наличия сильной корреляции между факторами приводит к неточным оценкам коэффициентов. Необходимо проводить более серьезный анализ признаков на наличие взаимосвязей.

В-третьих, *Dota 2* – это динамичная и сложная игра, ошибка всего одного игрока может привести к поражению команды, из-за этого нередко более слабые команды выигрывают сильных. Не всегда понятно, как и нужно ли такие специфические ситуации учитывать при построении моделей.

В результате разработанную модель можно использовать в качестве помощника для ставок на киберспортивные события и даже реализовать на ее основе коммерческий веб-продукт.



# 5. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА

В данной части работы рассмотрены функциональная модель разрабатываемой программной системы, архитектура, модель данных, структурное проектирование, описан выбор средств разработки.

5.1. Функциональная модель

На рис. 5.1 представлена диаграмма вариантов использования, отражающая функциональные требования к программному комплексу, т.е. как различные группы пользователей могут взаимодействовать с системой.

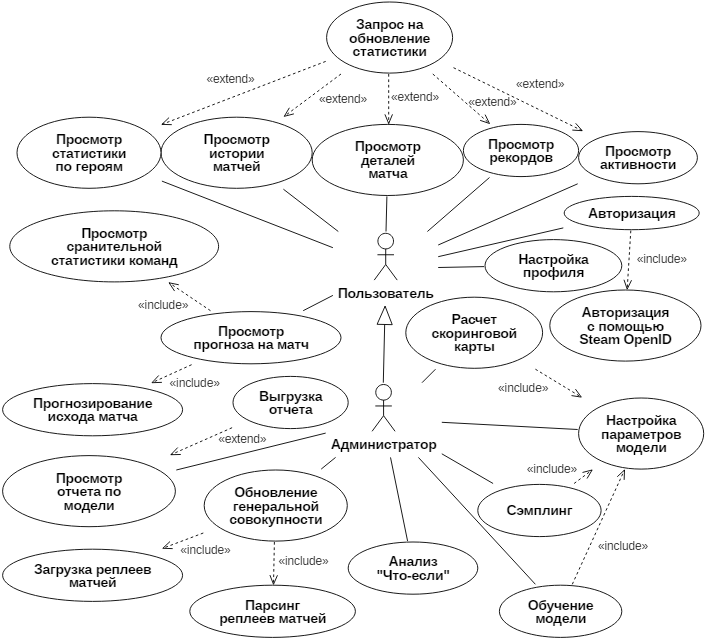


Рис 5.1. Диаграмма вариантов использования

Из диаграммы видно, что пользователь может просматривать различную персональную статистику, а также запрашивать прогноз результата определенного матча, а в ответ получать прогноз на матч и сравнительную статистику команд. Администратор может выполнять все действия, которые доступны пользователю, а также имеет возможность просматривать и выгружать отчет по модели, обновлять генеральную совокупность, производить сэмплинг, переобучать модель, обновлять скоринговую карту, работать с анализом «Что-если». На диаграмме можно видеть, что при обновлении генеральной совокупности обязательно происходит загрузка новых реплеев матчей и их парсинг, поэтому на стрелке указано отношение включения *“include”*.

С системой могут работать только авторизованные пользователи, а поскольку аккаунт *Dota 2* жестко привязан к игровой платформе *Steam*, авторизация реализована через серверы *Steam* с помощью механизма *Steam OpenID*. На диаграмме не указана регистрация, поскольку, если в системе авторизуется неизвестный пользователь, необходимые данные его *Steam*-аккаунта сохраняются в базе данных, что и является регистрацией.

5.2. Архитектура программной системы

Для реализации данного проекта была выбрана трехуровневая архитектура, состоящая из трех отдельных логических частей (рис. 5.2), каждая из которых физически размещена на разных серверах: веб-приложение (слой представления), веб-сервер (бизнес-слой), сервер БД (слой данных).

Веб-приложение взаимодействует с пользователем, представляет собой набор веб-страниц со скриптами в браузере и обменивается данными с веб-сервером по протоколу *HTTPS*. Данный уровень включает простейшую бизнес-логику: интерфейс авторизации, проверку вводимых значений, сортировку данных. В бизнес-слое сосредоточена большая часть бизнес-логики, включая запросы к серверу БД. Слой данных включает функциональность, связанную с доступом к данным. Веб-сервер и сервер БД находятся в локальной сети и обмениваются данными по протоколу *TCP* для того, чтобы обеспечить высокую скорость передачи данных. Также на рис. 5.2 можно видеть внешние источники данных, с которых загружаются файлы реплеев матчей и необходимые дополнительные данные. Трехуровневая архитектура имеет следующие преимущества [35]:

* уровни не зависят друг от друга, что позволяет производить обновления или изменения, не оказывая влияния на приложение в целом;
* уровни организовываются на основании развертывания слоев, поэтому масштабировать приложение довольно просто;
* изолированность уровней системы обеспечивает максимальную защищенность;
* уровни позволяют распределить нагрузку между серверами.

К недостаткам можно отнести:

* сложность развертывания и адмиинстрирования;
* высокие требования к производительности серверов;
* высокие требования к сети между серверами.

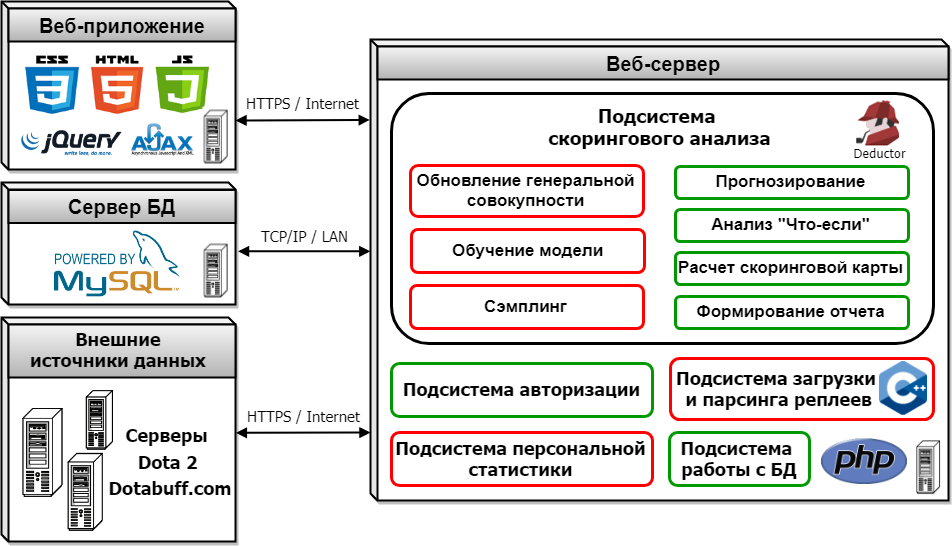


Рис. 5.2. Архитектура программной системы

Подсистема персональной статистики отвечает за предоставление пользователю его игровой статистики, ее обновление. Подсистема скорингового анализа отвечает за прогнозирование результатов матчей и обновление скоринговой модели. Подсистема работы с БД отвечает за извлечение и сохранение данных в базы данных. Подсистема загрузки и парсинга реплеев отвечает за загрузку файлов реплеев матчей с удаленных серверов и локальный их парсинг. В общем, подсистемы можно понимать как наборы определенных классов с их методами, реализующими требуемую функциональность и представленными в виде программных артефактов (скрипты, исполняемые файлы и т.д.).

Реализованная функциональность программного комплекса на рис. 5.2 выделена зеленым цветом.

5.3. Средства разработки

Для реализации программного комплекса были выбраны следующие средства разработки:

* Связка языка программирования *PHP* версии 7.2, СУБД *MySQL* версии 5.8 и *Apache HTTP*-сервер версии 2.4. *MySQL* – это одна из самых распространенных СУБД, а *Apache* – один из самых популярных *HTTP*-серверов. *PHP* – это простой язык, позволяет создавать динамические веб-страницы и отделять программную составляющую сайта от контента, имеет развитую поддержку баз данных, огромное количество библиотек. Поддержка сервера *MySQL* автоматически включается в поставку *PHP*.
* Локальный веб-сервер *Open Server* версии 5.2.2, который поставляется совместно с *PHP*, *MySQL* и *Apache*.
* Для разработки парсера реплеев матчей выбран язык программирования C*++* с поддержкой стандарта C++14, поскольку файлы реплеев весят десятки мегабайт и необходима высокая скорость обработки данных. тест
* Для разработки скоринговой модели используется программа *Deductor Studio Academic 5.3*, которая представляет собой аналитическую платформу для создания прикладных решений в области анализа данных. Реализованные в программе инструменты позволяют пройти все этапы разработки скоринговых моделей без необходимости программирования.
* Используемая среда разработки *Visual Studio Code* бесплатна, имеет встроенный локальный веб-сервер и множество различных подключаемых расширений, облегчающих разработку ПО.

5.4. Модель данных

На рис. 5.3 представлен фрагмент диаграммы «сущность-связь», описывающей физический уровень разработанной модели данных подсистемы скорингового анализа.

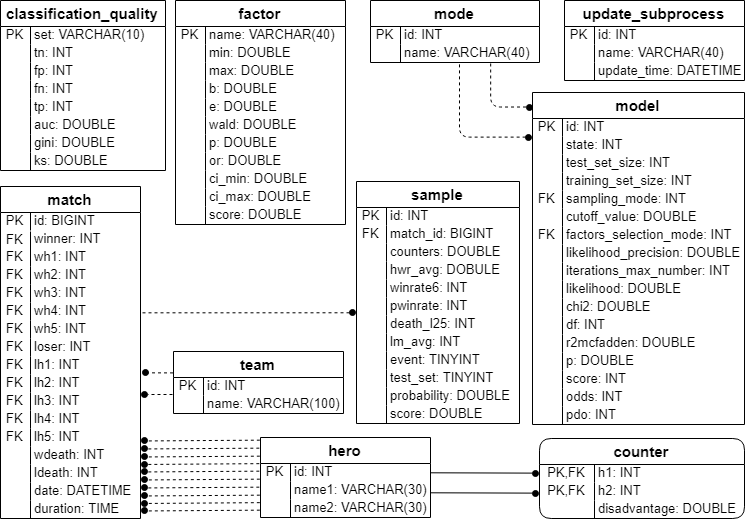


Рис. 5.3. Фрагмент диаграммы «сущность-связь». Физический уровень модели данных *подсистемы скорингового анализа*

Список имеющихся таблиц:

* таблица *team* описывает команду (идентификатор, название команды);
* таблица *hero* описывает героя (идентификатор, название героя в двух вариантах);
* таблица *match* описывает матч между двумя командами (идентификатор, идентификаторы команд и героев, число смертей каждой команды, продолжительность и дата матча);
* таблица *сounter* описывает эффективность одного героя против другого (идентификаторы героев, показатель эффективности);
* таблица *sample* описывает запись из выборки данных, используемых для обучения и тестирования модели (идентификатор, идентификатор матча, значения входных и выходной переменных, признак тестового множества, вероятность, скоринговый балл);
* таблица *factor* описывает коэффициент регрессии, соответствующий определенному фактору (название фактора, диапазон принимаемых значений, коэффициент регрессии, стандартная ошибка, коэффициент Вальда, значимость, отношение шансов, 95% доверительный интервал отношения шансов, скоринговый балл);
* таблица *classification\_quality* описывает результат классификации моделью (тип выборки данных, число истинно положительных, истинно отрицательных, ложно отрицательных, ложно положительных распознанных наблюдений, показатели *AUC*, *GINI*, *KS*);
* таблица *update\_subprocess* описывает подпроцесс процесса обновления модели (идентификатор, название, дата последнего обновления);
* таблица *mode* описывает алгоритм обработки данных (идентификатор, название алгоритма);
* таблица *model* описывает модель (идентификатор, состояние модели, размеры обучающей и контрольной выборок данных, идентификатор метода сэмплинга, порог отсечения, идентификатор метода отбора переменных, точность функции оценки, максимальное число итераций, значение логарифма функции правдоподобия, хи-квадрат, число степеней свободы, коэффициент детерминации МакФаддена, значимость, параметры масштабирования скоринговой карты).

5.5. Структурное проектирование

На рис. 5.4 представлен фрагмент диаграммы классов, демонстрирующей структуру иерархии основных классов системы, связи между ними, их атрибуты и методы.

Класс *Client* описывает пользователя, взаимодействующего с системой, инкапсулирует механизм сессий из языка *PHP*, который позволяет сохранять информацию о пользователях между запросами страниц. Классы *PersonalStatisticsSubsystem* и *ScoringSubsytem* являются классами-контроллерами соответствующих подсистем, обрабатывают запросы от клиентов, взаимодействуют с другими классами и возвращают результат. Класс *ScoringModel* – это ядро скоринговой подсистемы, содержит методы для обучения модели, расчета скоринговой карты и другие. Использует программу *Deductor Studio* для реализации сэмплинга и обучения модели. Очевидно, что класс *Authorization* отвечает за авторизацию в системе. Класс *ReplayParser* содержит методы для загрузки и парсинга файлов реплеев, при этом для парсинга используется исполняемый файл, разработанный на языке C++. В результате парсинга создается объект класса *Replay*, не имеющий методов и содержащий необходимые данные о матче. Классы *DBSS*, *DBPSS*, *DBUsers* содержат методы для работы с базами данных.

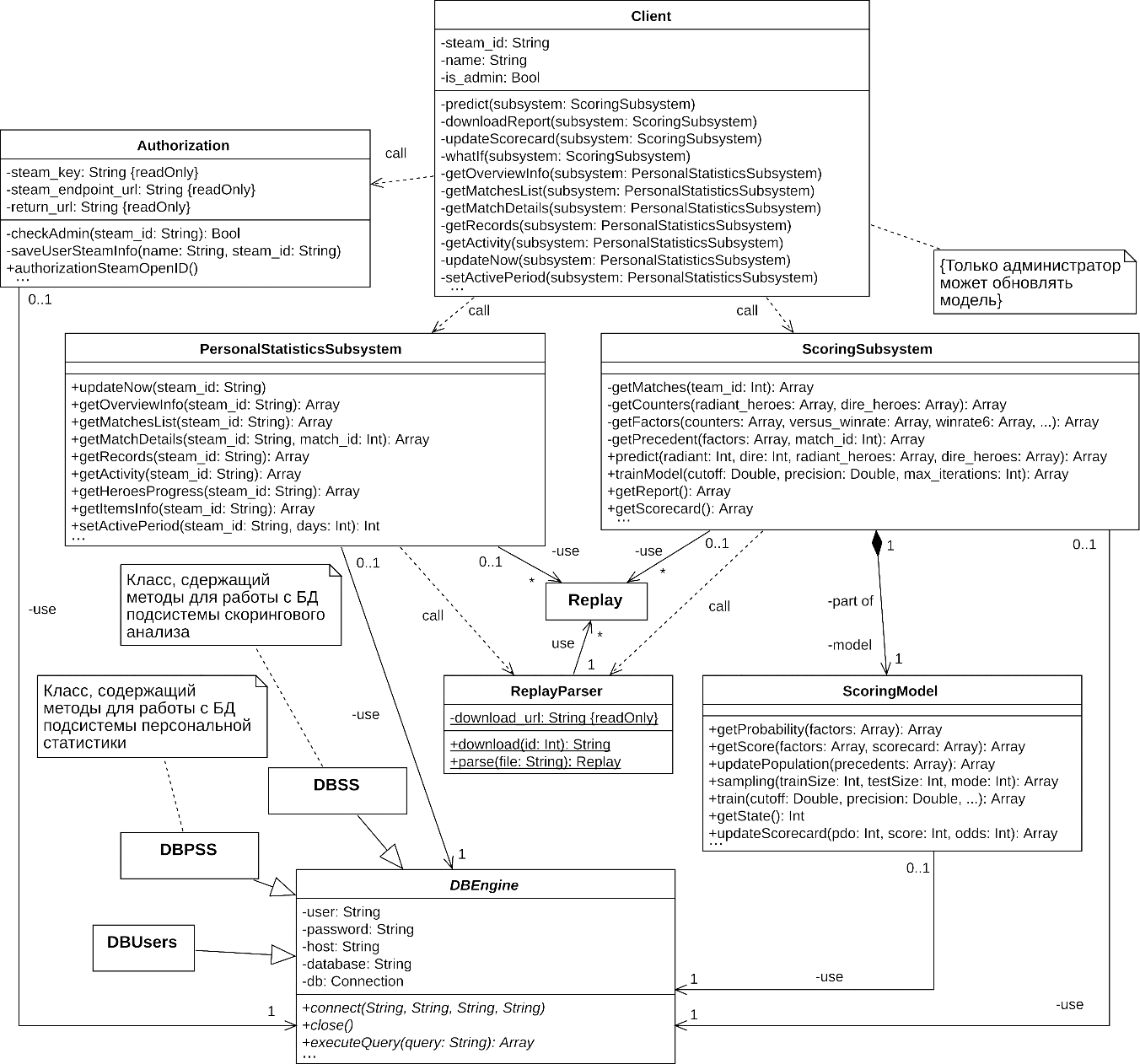


Рис. 5.4. Фрагмент диаграммы классов

Следует отметить, что в *PHP* для реализации словарей, списков, массивов повсеместно используется тип *Array*.

На рис. 5.5 представлена диаграмма последовательности для процесса прогнозирования исхода матча. Данная диаграмма демонстрирует взаимодействие группы объектов между собой в процессе прогнозирования исхода матча, обеспечивает наглядное представление порядка передачи сообщений. Тонкие вертикальные прямоугольники соответствуют промежутку времени, когда объект активен, например, подсистема авторизации выполняет определенные действия, только когда пользователь авторизуется в системе.

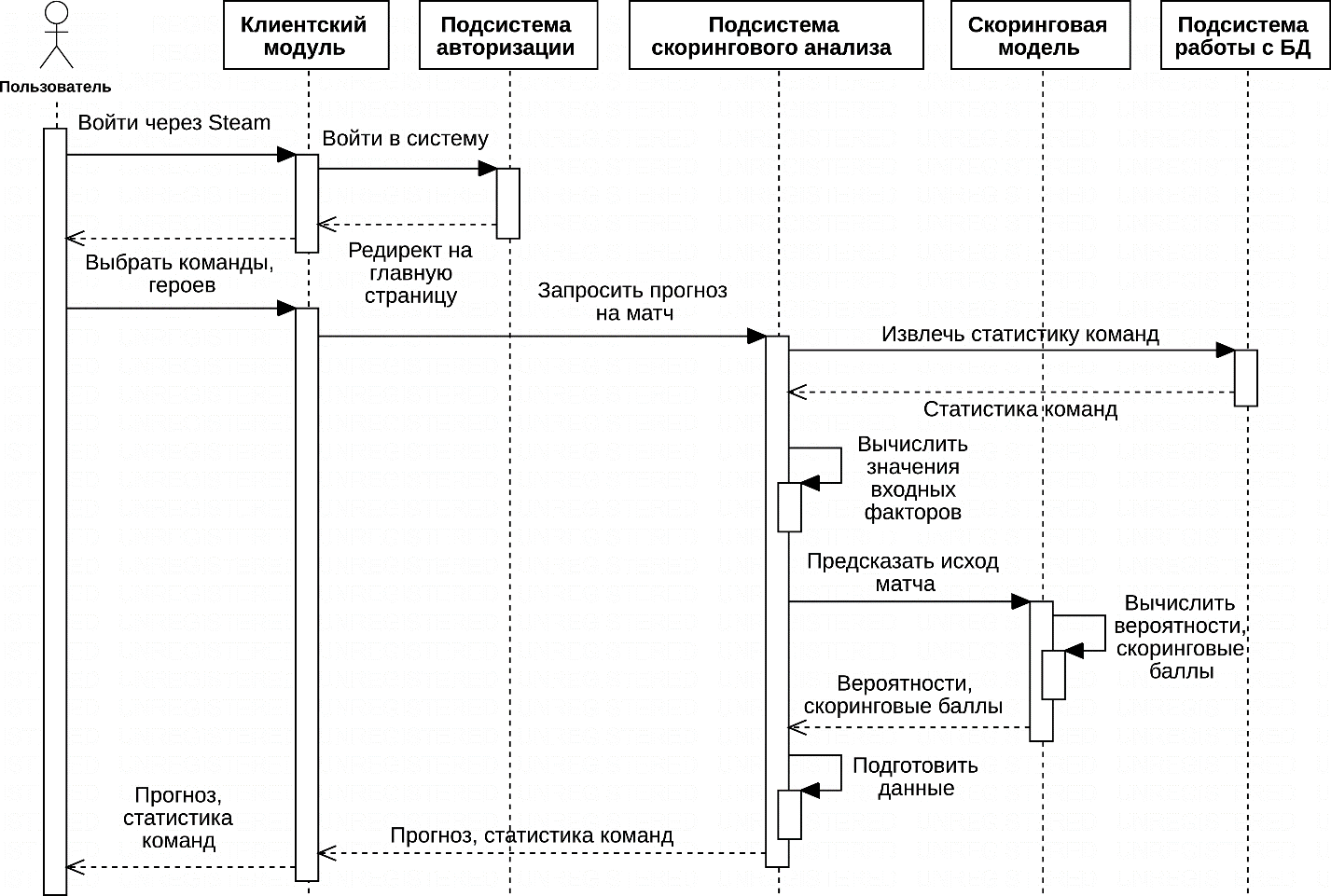


Рис. 5.5. Диаграмма последовательности. Процесс прогнозирования исхода матча

На рис. 5.6 представлена диаграмма последовательности для процесса обновления скоринговой модели, включающего обновление генеральной совокупности, сэмплинг, обучение модели и расчет скоринговой карты. Каждый этап выполняется последовательно, например, нельзя переобучить модель сразу после обновления генеральной совокупности, не запустив сэмплинг, поскольку количество записей, используемых для обучения модели, изменится после обновления данных. Защита от таких ситуаций реализована в программном коде и на стороне клиента, и на сервере, поэтому похожие сценарии работы с системой не могут привести к некорректному состоянию скоринговой модели. В то же время, например, если произведен сэмплинг, то переобучать модель можно множество раз, задавая различные входные параметры.

На рис. 5.7 представлена диаграмма развертывания, демонстрирующая физическое расположение системы и используемое программное обеспечение. Следует отметить, что на диаграмме изображены только некоторые из имеющихся *php*-скриптов. Поскольку программа *Deductor Studio* работает только на ОС *Windows*, поэтому используется серверная ОС *Windows Server*. Из *php*-скриптов с заданными параметрами командной строки запускаются исполняемые файлы *Deductor.exe* и *Parser.exe* и работают в пакетном режиме. Для работы с базами данных используется обновленная версия драйвера *PHP MySQL* – *PHP* *MySQLi Extension*, которая предоставляет различные улучшения в работе с базами данных, в частности, объектно-ориентированный интерфейс и поддержку транзакций.

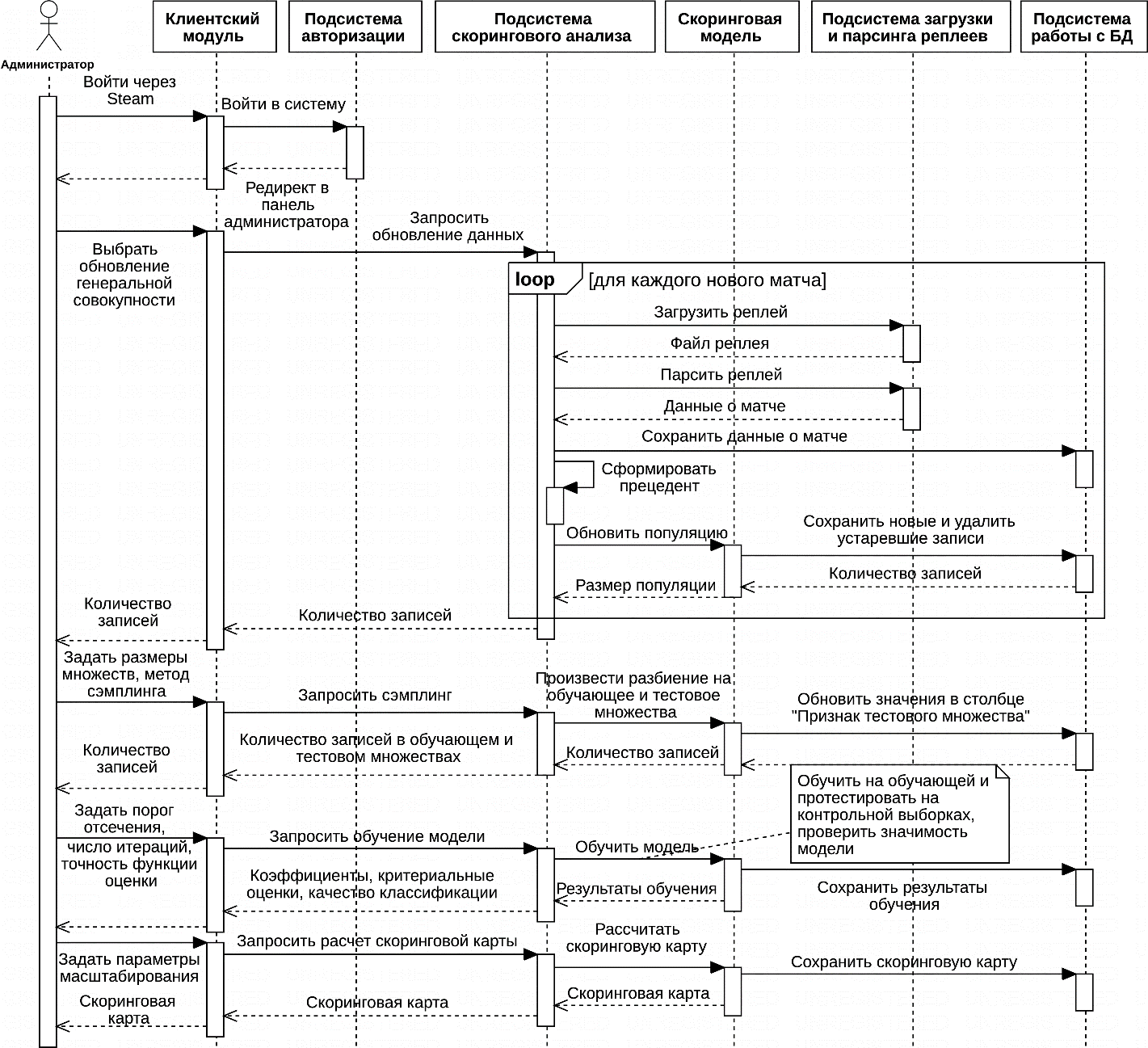


Рис. 5.6. Диаграмма последовательности. Процесс обновления скоринговой модели

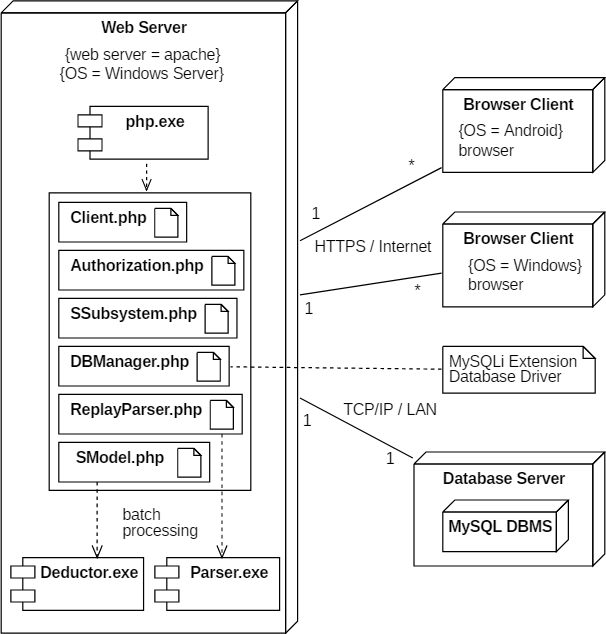


Рис. 5.7. Диаграмма развертывания

# 6. ТЕСТИРОВАНИЕ ПРОГРАММНОГО КОМПЛЕКСА

Для проверки соответствия программного комплекса заявленным требованиям, а также для выявления различного рода ошибок и нестандартных ситуаций поведения программной системы необходимо провести тестирование.

При тестировании программного продукта за основу был взят метод «серого ящика». Предполагается, что специалист по тестированию имеет доступ к внутренней структуре и алгоритмам работы ПО, но само тестирование проводится методом «черного ящика», т.е. с позиции пользователя.



## 6.1. План испытаний

Запланированы следующие этапы процесса тестирования:

1) Функциональное тестирование.

a) Тестирование авторизации.

b) Тестирование сервиса прогнозирования исходов матчей.

c) Тестирование анализа «Что-если».

2) Тестирование скоринговой модели.

a) Сравнение моделей, используя при обучении от 1 до 6 из отобранных в финальную модель входных факторов в порядке уменьшения их значимости.

3) Нагрузочное тестирование сервиса прогнозирования исходов матчей.

4) Тестирование верстки.

## 6.2. Функциональное тестирование

### 6.2.1. Тестирование авторизации

Для тестирования авторизации использовались 3 различных *Steam*-аккаунта. Во всех трех случаях после нажатия на кнопку *“Sign in to continue”* произошло перенаправление на страницу авторизации веб-ресурса *Steamcommunity.com*, после успешной авторизации на данном сайте произошло перенаправление на исходный сайт под правами зарегистрированного пользователя (рис. 6.1).

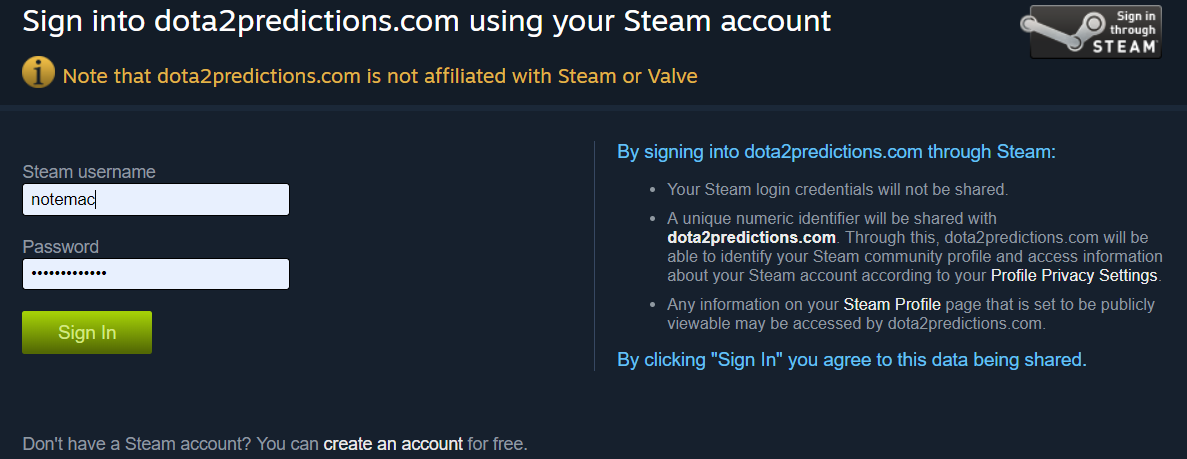


Рис. 6.1. Авторизация на сайте через Steam

### 6.2.2. Тестирование сервиса прогнозирования исходов матчей

На странице сервиса была произведена следующая последовательность действий: выбраны две команды из доступного списка команд, выбраны 10 героев из доступного списка героев, нажата кнопка *“Predict”*. Через несколько секунд корректно отобразились вероятности побед и сравнительная статистика для выбранных команд (рис. 6.2).

При повторении последовательности действий, указанных выше, но при выборе только одной команды (или не выбрав ни одной) или менее 10 героев, система выдала сообщение об ошибке (рис. 6.3).

Данные сценарии работы с системой были произведены более 100 раз, выбирая каждый раз различные комбинации и количество команд и героев, ошибки обнаружены не были.

### 6.2.3. Тестирование анализа «Что-если»

Для проверки правильности расчетов, получаемых с помощью анализа «Что-если», использовались девять различных прецедентов. В табл. 6.1 представлены результаты тестирования, в которой столбцы *PWI* и *SCOREWI* содержат вероятности и скоринговые баллы, рассчитанные с помощью анализа «Что-если», а столбцы *P* и *SCORE* содержат значения, рассчитанные вручную. Во всех случаях расчеты оказались правильными.



Рис. 6.2. Сервис прогнозирования исходов матчей. Вероятности и сравнительная статистика команд

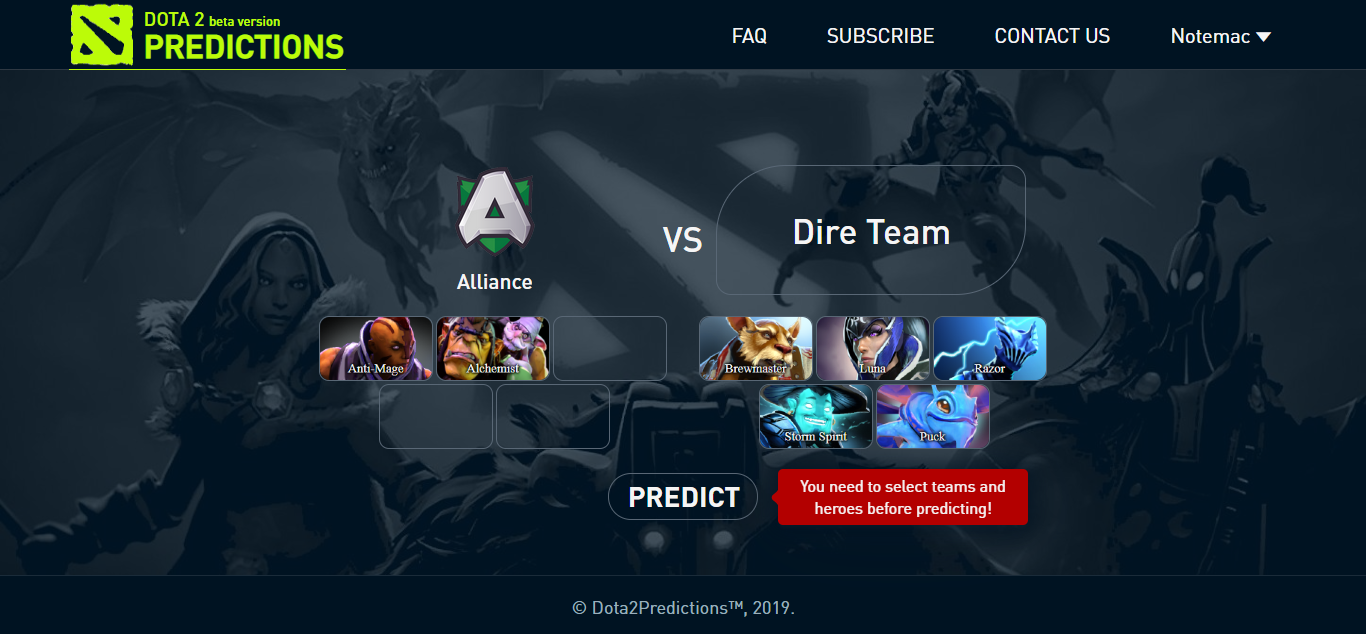


Рис. 6.3. Сервис прогнозирования исходов матчей. Сообщение об ошибке

Таблица 6.1

Результаты тестирования анализа «Что-если»

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***counters*** | ***pwinrate*** | ***winrate12*** | ***hwr\_avg*** | ***death\_l25*** | ***lm\_avg*** | ***PWI*** | ***SCOREWI*** | ***P*** | ***SCORE*** |
| -0,3 | 50 | 1 | 1 | -1 | 10 | 0,68 | 509,43 | 0,68 | 509,43 |
| -0,07 | 45 | 3 | -11,08 | 9 | -29 | 0,53 | 490,48 | 0,53 | 490,48 |
| 0,3 | 51 | 14 | 20 | 13 | -180 | 0,65 | 505,08 | 0,65 | 505,08 |
| -0,25 | 14 | -7 | -5,326 | 20 | 44 | 0,54 | 491,49 | 0,54 | 491,49 |
| 0 | 46 | 10 | 8,14 | 12 | -22 | 0,7 | 511,44 | 0,7 | 511,44 |
| 0,58 | 46 | -9 | -17 | -11 | 4 | 0,09 | 419,17 | 0,09 | 419,17 |
| 0,092 | 51 | -18 | -18,01 | -15 | -183 | 0,07 | 412,1 | 0,07 | 412,1 |
| -0,22 | 54 | 2 | 3,1 | -2 | -50 | 0,68 | 509,43 | 0,68 | 509,43 |
| 0,04 | 48 | 16 | 18 | 12 | -21 | 0,84 | 534,74 | 0,84 | 534,74 |

На рис. 6.4 представлен результат работы анализа «Что-если» для последнего прецедента из табл. 6.1.

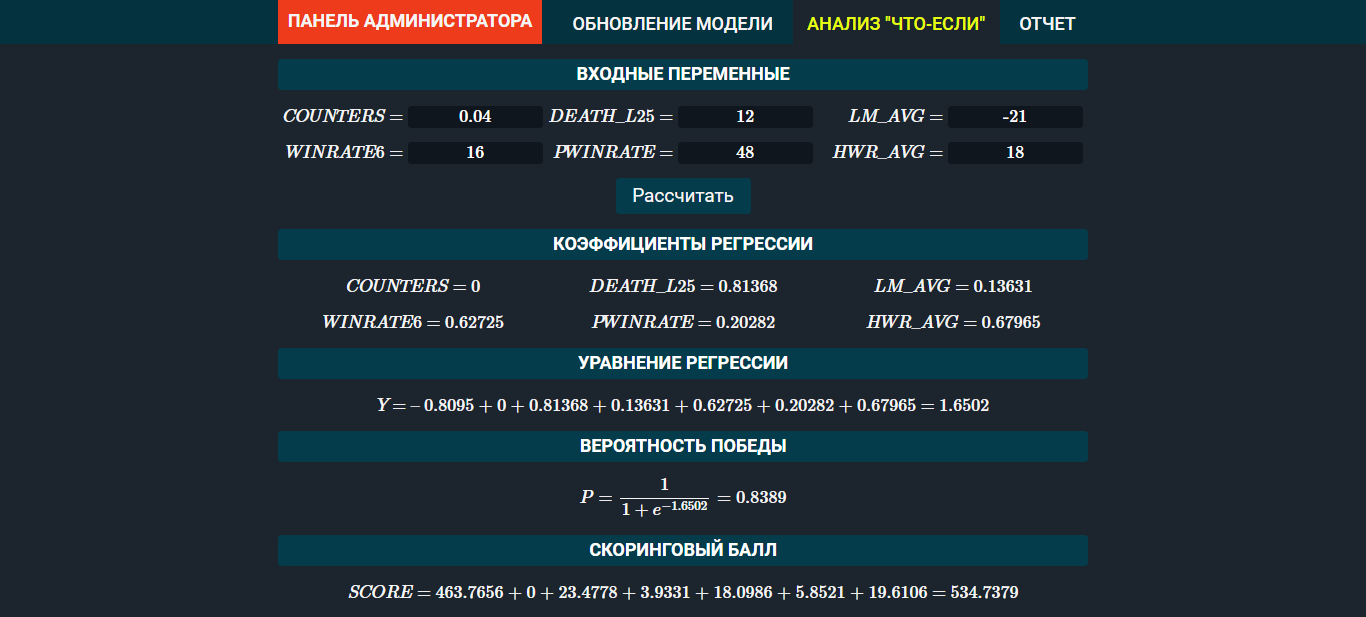


Рис. 6.4. Анализ «Что-если»

## 6.3. Тестирование скоринговой модели

### 6.3.1. Сравнение моделей с разным числом входных факторов

При обучении использовались следующие признаки, перечисленные в порядке уменьшения значимости: *COUNTERS*, *PWINRATE*, *WINRATE12*, *HWR\_AVG*, *DEATH\_L25*, *LM\_AVG*.

Для сравнения моделей выбраны следующие показатели: *OSR* – доля правильно классифицированных наблюдений, *AUC* – оценка площади под *ROC*-кривой [32], *AIC* – информационный критерий Акаике [36]. В табл. 6.2 представлены значения данных критериев для каждой модели, рассчитанные в программе *Deductor Studio*.

Таблица 6.2

Значения показателей OSR, AUC и AIC для моделей с разным числом входных факторов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Количество факторов** | ***AUCобуч*** | ***AUCтест*** | ***OSRобуч, %*** | ***OSRтест, %*** | ***AIC*** |
| 1 | 0,68 | 0,68 | 62,7 | 63,2 | 31802 |
| 2 | 0,75 | 0,75 | 67,7 | 67,2 | 28867 |
| 3 | 0,78 | 0,78 | 68,7 | 67,8 | 27518 |
| 4 | 0,78 | 0,78 | 68,8 | 67,9 | 27276 |
| 5 | 0,79 | 0,79 | 69,3 | 68,6 | 26821 |
| 6 | 0,79 | 0,79 | 69,7 | 68,7 | 26677 |

В табл. 6.2 видно, что с увеличением числа входных факторов увеличивается значение показателя *AUC*. Чем оно больше, тем выше предсказательная способность модели, что подтверждается данными из 4 и 5 столбцов. Критерий *AIC* основан на оценке потери информации при уменьшении числа параметров модели, за увеличение числа параметров модель «штрафуется». Предпочтение отдается модели, для которой значения критерия окажется минимальным. Согласно последнему столбцу в табл. 6.2, чем больше входных факторов, тем эффективнее модель. Полученные результаты также свидетельствуют, что только лишь на основании информации о том, насколько одни герои сильнее других, можно добиться прогнозирующей точности более 60%.

## 6.4. Нагрузочное тестирование сервиса прогнозирования исходов матчей

Сервис прогнозирования исходов матчей рассчитан на 200 одновременно работающих пользователей. При числе запросов в секунду, равном 15, время выполнения запросов не должно превышать 3 секунд.

На рис. 6.5 и рис. 6.6 представлены результаты тестирования, проводимого в течение 2 минут с эмуляцией работы 200 пользователей. Согласно графикам, максимальное число запросов в секунду было 18, а максимально время выполнения запросов составило 2,7 секунд. Среднее число запросов в секунду – 17,4, среднее время ответа от сервера – 2 секунды.

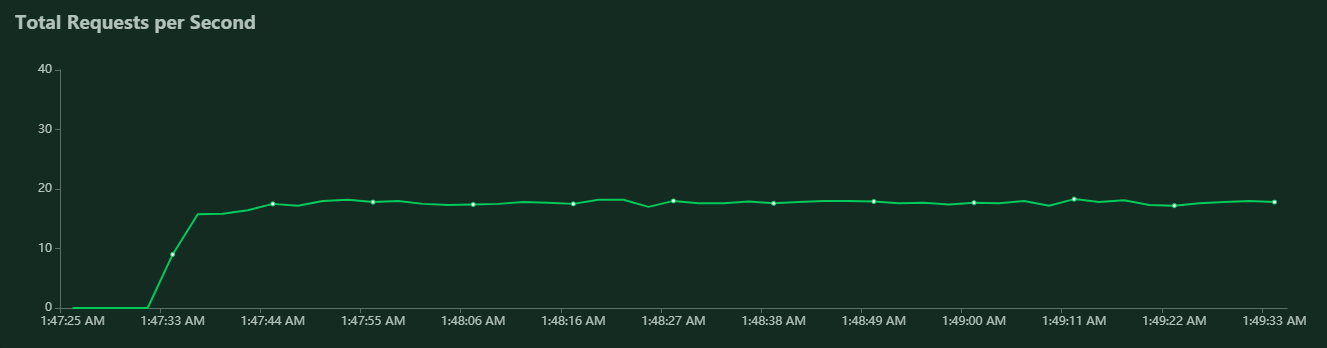


Рис. 6.5. Количество запросов в секунду

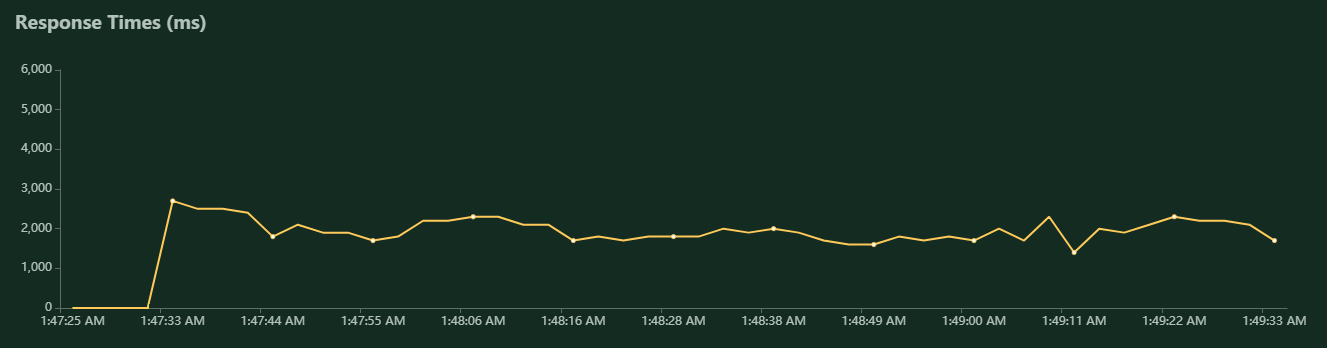


Рис. 6.6. Время выполнения запросов

По полученным результатам можно сделать вывод, что нагрузка в 200 пользователей сайт не нагружает и даже имеется небольшой запас производительности.

## 6.5. Тестирование верстки

На рис. 6.7 представлена страница управления скоринговой моделью, открытая в трех браузерах последних версий при увеличении в 150% (в порядке сверху вниз и слева направо): *Google Chrome*, *Mozilla Firefox*, *Opera*.



Рис. 6.7. Страница управления скоринговой моделью

Страница отображается корректно во всех браузерах, следовательно никаких исправлений вносить не нужно, верстка разработана правильно.

# 7. ОРГАНИЗАЦИОННАЯ ЧАСТЬ

Появление компьютеров в нашей жизни изменило характер труда, что потребовало комплексного решения проблем эргономики, гигиены и организации труда, регламентации режимов труда и отдыха.

При работе за компьютером человек подвергается воздействию ряда опасных и вредных производственных факторов:

* нарушение микроклимата помещения;
* повышенный уровень шума на рабочем месте;
* повышенный уровень статического электричества;
* повышенный уровень электромагнитных излучений;
* повышенная напряженность электрического и магнитного полей;
* отсутствие или недостаток естественного света;
* недостаточная освещенность рабочей зоны.

Для уменьшения негативных последствий работы за компьютером следует выбирать рациональные режимы труда и отдыха, использовать защитные средства, осуществлять комплексные оздоровительно-профилактические мероприятия. Безопасные условия труда при работе с ПЭВМ регламентируют СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к ПЭВМ и организации работы» [37].

Несоблюдение требований безопасности приводит к тому, что через некоторый период времени здоровье человека подвергается опасности. Он начинает испытывать усталость, раздражительность, головные боли, дискомфорт в области спины и шеи, и это далеко не все последствия, которые ожидают человека, несоблюдающего требования безопасности.

## 7.1. Организация режима труда и отдыха

Для обеспечения оптимальной работоспособности и сохранения здоровья профессиональных пользователей персональных компьютеров на протяжении рабочей смены должны устанавливаться регламентированные перерывы. Время регламентированных перерывов в течение рабочей смены следует устанавливать в зависимости от ее продолжительности, вида и категории трудовой деятельности.

Продолжительность непрерывной работы перед монитором без регламентированного перерыва не должна превышать 1 час. Поскольку характер работы программистов требует постоянного взаимодействия с монитором персонального компьютера с напряжением внимания и сосредоточенности, при исключении возможности периодического переключения на другие виды трудовой деятельности, рекомендуется организация перерывов на 10-15 минут через каждые 45-60 минут работы.

Во время регламентированных перерывов для снижения нервно-эмоционального напряжения, утомления органов зрения целесообразно выполнять различные комплексы упражнений для глаз, также физкультминутки, способствующие снятию локального утомления.

Согласно СанПиН 2.2.2/2.4.1340-03 экран монитора должен находиться от глаз пользователя на расстоянии 600-700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов [37]. С целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

## 7.2. Электробезопасность

Электрический ток представляет собой скрытый тип опасности, поскольку его трудно определить в токо- и нетоковедущих частях оборудования, которые являются хорошими проводниками электричества. Смертельно опасным для жизни человека считают ток, величина которого превышает 0,05 А. С целью предупреждения поражений электрическим током к работе должны допускаться только лица, хорошо изучившие основные правила по технике безопасности.

В соответствии с правилами электробезопасности в помещении должен осуществляться постоянный контроль состояния электропроводки, предохранительных щитов, шнуров, с помощью которых включаются в электросеть компьютеры, осветительные приборы, другие электроприборы.

ЭВМ представляют для человека большую потенциальную опасность, так как в процессе эксплуатации или проведении профилактических работ человек может коснуться частей, находящихся под напряжением в результате повреждения изоляции. Исключительно важное значение для предотвращения электротравматизма имеет правильная организация обслуживания действующих электроустановок.

Разрядные токи статического электричества чаще всего возникают при прикосновении к любому из элементов ЭВМ. Такие разряды опасности для человека не представляют, но кроме неприятных ощущений они могут привести к выходу из строя ЭВМ. Для снижения величины возникающих зарядов статического электричества покрытие полов следует выполнять из однослойного поливинилхлоридного антистатического линолеума. К общим мерам защиты от статического электричества можно отнести общие и местное увлажнение воздуха.

## 7.3. Синдром запястного канала

Любое статическое положение, в котором человек находится долго в максимальной концентрации, рано иди поздно приведет к болям. Программисты же почти все свое рабочее время проводят в сидячем положении за компьютером. Активное и долговременное использование клавиатуры и мыши в неправильной позе может привести к возникновению синдрома запястного канала.

Синдром запястного канала – неврологическое заболевание, проявляющееся длительной болью и онемением пальцев кисти. Основные симптомы данного заболевания – мурашки, покалывание, онемение первого, второго и третьего пальцев рук. Один из методов диагностики – это тест Фалена: необходимо максимально согнуть кисть в лучезапястном суставе на 1 минуту. Если появляются мурашки, усиливается онемение, то тест положительный и следует идти к врачу.

Для того чтобы снизить риск возникновения данного заболевания, работникам следует придерживаться следующих общих рекомендаций:

* необходимо иногда менять позицию, например, во время перерывов в работе;
* сделать себе удобное рабочее место;
* подобрать такую компьютерную мышь, чтобы она хорошо лежала в руке и было комфортно с ней работать;
* кисть при работе с клавиатурой не должна быть разогнута на 20 и более градусов по отношению к предплечью.

## 7.4. Требования к освещенности

Освещение влияет на физическое и психическое состояние человека. При недостаточной освещенности в процессе работы развивается утомление зрения, понижается общая работоспособность и производительность труда.

Согласно СанПиН 2.2.2/2.4.1340-03 рабочие столы следует размещать таким образом, чтобы мониторы персональных компьютеров были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева [37]. Освещенность на поверхности рабочего стола должна быть 300-500 лк. Допускается установка светильников местного освещения, но с таким условием, чтобы оно не создавало бликов на поверхности экрана и не увеличивало освещенность экрана более чем на 300 лк.

Общее освещение следует выполнить в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения работников при рядном расположении персональных компьютеров. При периметральном расположении компьютеров линии светильников следует расположить локализовано над рабочим столом, ближе к его переднему краю, обращенному к человеку.

В качестве источников света при искусственном освещении должны применяться преимущественно люминесцентные лампы типа ЛД. Для обеспечения требуемых значений освещенности в помещении необходимо проводить чистку стекол оконных рам и светильников не реже двух раз в год и своевременную замену перегоревших ламп.

## 7.5. Пожарная безопасность

Согласно ГОСТ 12.1.033-81 пожарная безопасность – состояние объекта, при котором с установленной вероятностью исключается возможность возникновения и развития пожара и воздействия на людей опасных факторов пожара, а также обеспечивается защита материальных ценностей [38]. Пожарная безопасность обеспечивается совокупностью организационных мероприятий и технических средств, направленных на предотвращение воздействия на людей опасных факторов пожара и ограничение материального ущерба от него.

Характерная особенность офисов, в которых работают группы программистов, – небольшие площади помещений, в которых может находиться множество дорогостоящей электронной техники. Поэтому пожары в таких местах представляют особую опасность, так как сопряжены с большими материальными потерями.

К основным причинам возникновения пожара в таком помещении можно отнести возгорание электронного оборудования и неисправность электрической проводки. Источниками зажигания могут быть электронные схемы от ЭВМ, приборы, применяемые для технического обслуживания, устройства электропитания, кондиционирования воздуха, где в результате различных нарушений образуются перегретые элементы, электрические искры и дуги, способные вызвать загорания горючих компонентов (мебель, полы, изоляция кабелей и др.).

В современных ЭВМ очень высокая плотность размещения элементов электронных схем. В непосредственной близости друг от друга располагаются соединительные провода, кабели. При протекании по ним электрического тока выделяется значительное количество теплоты. При этом возможно оплавление изоляции.

Согласно ГОСТ 12.1.033-81 пожарная профилактика – это комплекс организационных и технических мероприятий, направленных на обеспечение безопасности людей, на предотвращение пожара, ограничение его распространения, а также на создание условий для успешного тушения пожара [38].

Одной из наиболее важных задач пожарной профилактики является защита помещений от разрушений и обеспечение их достаточной прочности в условиях воздействия высоких температур при пожаре. Учитывая высокую стоимость электронного оборудования и категорию его пожарной опасности, здания, в которых предусмотрено размещение ЭВМ, должны быть 1 и 2 степени огнестойкости согласно СНиП 21-01-97 «Пожарная безопасность зданий и сооружений» [39].

Для того чтобы предупредить возникновение пожара в помещении, необходимо принять следующие меры:

* укомплектовать помещение необходимыми средствами пожаротушения и пожарным оборудованием;
* не перегружать электросеть:
* использовать исправное электронное оборудование;
* по окончании работы выключать электронную технику;
* следовать инструкции по технике безопасности при работе за различным оборудованием.

Для тушения пожаров на начальных стадиях широко применяются огнетушители. В данном случае рекомендуется применять углекислотные огнетушители, достоинством которых является высокая эффективность тушения пожара, сохранность электронного оборудования, диэлектрические свойства углекислого газа, что позволяет использовать эти огнетушители даже в том случае, когда не удается обесточить электроустановку сразу.

Также следует отметить, что в помещении обязательно должен находиться на видном месте «План эвакуации людей при пожаре», регламентирующий действия персонала в случае возникновения пожара и указывающий места расположения пожарной техники.

## 7.6. Расчет освещенности

Исходные данные:

* тип лампы – люминесцентная, тип светильника – ЛД;
* длина помещения , ширина помещения , площадь помещения ;
* требуемая освещенность в помещении ;
* высота подвеса светильника .

Согласно типу светильника выбирается коэффициент , определяющий соотношение максимального расстояния между светильниками и высотой их подвеса над рабочей поверхностью , которое обеспечит равномерность освещения в помещении. Тогда максимально допустимое расстояние между светильниками будет равно:

.

Расстояние от стены до первого ряда светильников составит:

.

Общее число рядов светильников (по ширине помещения) будет равно:

и число светильников в ряду (по длине помещения):

Общее расчетное минимальное количество светильников, которое необходимо разместить в помещении:

По площади помещения и высоте подвески светильника определяется показатель помещения:

Ориентировочное значение коэффициента отражения потолка помещения, для которого рассчитывается осветительная установка, принято равным , стен – и полов – .

В зависимости от типа светильника и вида лампы по показателю помещения и коэффициентам отражения потолка, стен и полов определяется коэффициент использования светового потока .

Коэффициент запаса, учитывающий снижение уровня освещенности из-за неблагоприятных условий эксплуатации осветительной установки: наличия дыма, копоти, пыли, повышенной концентрации химических веществ и т. д.; из-за старения и выхода из строя ламп, будет равен .

Коэффициент отношения средней освещенности к максимальной, характеризующий неравномерность освещенности, для люминесцентных ламп принимается равным .

Требуемый световой поток одной лампы составит:

где – количество источников света в светильнике.

По рассчитанному световому потоку лампы подбирается стандартная лампа со световым потоком , значение которого близко к значению (желательно в пределах от -10 до +20%). Затем рассчитывается число светильников, необходимых для обеспечения требуемой освещенности, и мощность проектируемой системы освещения. Для обеспечения минимального потребления электроэнергии осветительной установкой также необходимо рассчитать количество светильников, назначая лампы с меньшей мощностью:

(лм);

;

;

Результаты расчетов параметров осветительных установок приведены в табл. 7.1. На рис. 7.1 представлена схема размещения светильников в помещении.

Таблица 7.1

Результаты расчетов параметров осветительной установки

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Тип лампы** | **Световой поток лампы Ф, лм** | **Количество светильников** | | **Отклонение *n*пр от *n*расч, %** | **Полная мощность *N*, Вт** |
| **Расчетное *n*расч** | **Принятое *n*пр** |
| 1 | ЛД-65 | 3390 | 8,03 | 8 | -0,37 | 1040 |
| 2 | ЛД-40 | 2225 | 12,24 | 12 | -1,96 | 960 |
| 3 | ЛД-30 | 1560 | 17,45 | 18 | 3,15 | 1080 |

Вывод: вариант осветительной установки с типом ламп ЛД-40 является оптимальным, так как значение её полной потребляемой мощности меньше, чем у осветительных установок с типами ламп ЛД-65 и ЛД-30.

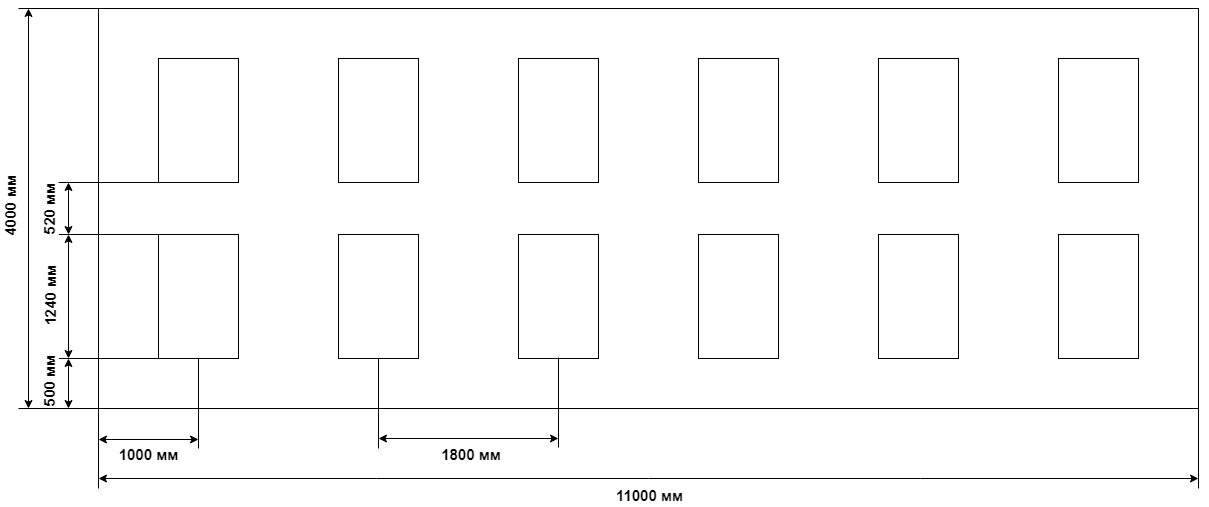


Рис. 7.1. Схема размещения светильников

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы была изучена предметная область, а именно процесс прогнозирования исхода матча между двумя командами в компьютерной игре *Dota 2*.

Произведен сравнительный анализ трех программных аналогов, также автор ознакомился с различными подходами и алгоритмами, которые могут применяться при реализации модели, прогнозирующей исход матча в *Dota 2*.

В техническом задании были определены основные требования к программному комплексу.

Далее была разработана скоринговая модель, максимальная общая точность прогнозирования которой составила 69,6% или 21669 правильно классифицированных из 31134 записей. Согласно техническому заданию разработанную модель можно считать успешной, к тому же точность оказалась не хуже некоторых рассмотренных аналогичных моделей, а почти по всем критериям значимости и качества модель можно считать хорошей.

На этапе проектирования была разработана и описана функциональная модель программного комплекса, отражающая как пользователи могут взаимодействовать с системой, архитектура программной системы, модель данных, диаграмма классов, демонстрирующая структуру иерархии основных классов системы, связи между ними, их атрибуты и методы, две диаграммы последовательности для процессов прогнозирования исхода матча и обновления скоринговой модели, показывающие, как объекты системы взаимодействуют между собой, диаграмма развертывания, демонстрирующая физическое расположение системы и используемое программное обеспечение, описаны используемые для разработки программные средства.

На этапе тестирования были проведены тестирование программного комплекса на соответствие функциональным требованиям, нагрузочное тестирование сервиса прогнозирования исходов матчей, тестирование верстки, сравнение моделей с разным числом входных факторов.

В экономической части были проведены расчеты затрат на создание программного комплекса.

В организационной части были даны общие рекомендации по охране труда и технике безопасности.

В результате выполнения работы спроектирована система интеллектуального анализа игровой статистики и частично реализована подсистема скорингового анализа в виде веб-приложения, а именно разработаны скоринговая модель, подсистемы работы с БД и авторизации, сервис прогнозирования исходов матчей, расчет скоринговой карты, анализ «Что-если», просмотр и выгрузка отчета по модели.

План дальнейшего развития и модернизации программного комплекса:

1) Полностью разработать программный комплекс.

2) Попытаться добиться повышения прогнозирующей точности модели за счет добавления новых входных признаков (например, учитывать стратегию команд на основе выбранных героев в матче).

3) Если в реальных условиях точность прогнозов сервиса прогнозирования исходов матчей будет сопоставима с результатами, полученными при тестировании модели, попытаться его монетизировать за счет платной подписки. Первый вариант: ежемесячная подписка за 1 доллар будет включать только базовую статистику по командам, подписка за 2 доллара – расширенную статистику. Второй вариант: подписка за 1 доллар позволит запросить 40 прогнозов, за 2 доллара – 80 и т.д. Можно придумать множество вариантов монетизации данного сервиса.

4) Букмекерские конторы предлагают делать ставки не только на исходы матчей, но и на другие события, например, сколько минут будет длиться матч или какая будет итоговая разница в счете. В планах добавить функциональность для анализа таких событий.

# СПИСОК ЛИТЕРАТУРЫ

1. Киберспортивный турнир The International 2018 по Dota 2 [Электронный ресурс] / Cybersport.ru – портал о киберспорте, новости, репортажи, турниры, расписание матчей, рейтинги команд и игроков. – Элекрон. дан., 2019. – Режим доступа: <https://www.cybersport.ru/base/tournaments/the-international-2018>.
2. STEAMCHARTS [Электронный ресурс] / Dota 2 – Steam Charts. – Электрон. дан., 2019. – Режим доступа: <https://www.steamcharts.com/app/570>.
3. О признании и включении во Всероссийский реестр видов спорта спортивных дисциплин, видов спорта и внесении изменений во Всероссийский реестр видов спорта: Приказ от 29 апреля 2016 г. № 470. – М.: Министерство спорта РФ, 2016. – 9 с.
4. Киберспортсмен – официально признанная должность в Китае [Электронный ресурс] / SPORTBOX – новости спорта, спортивная аналитика. – Электрон. дан., 2019. – Режим доступа: <https://news.sportbox.ru/Vidy_sporta/Kibersport/spbnews_NI1000375_Kibersportsmen__oficialno_priznannaja_dolzhnost_v_Kitaje>.
5. Dota 2 – Путь на The International [Электронный ресурс] / Dota 2 – Play for Free. – Элекрон. дан., 2019. – Режим доступа: <https://www.dota2.com/procircuit>.
6. DOTABUFF [Электронный ресурс] / DOTABUFF – Dota 2 Statistics. – Электрон. дан., 2019. – Режим доступа: <https://www.dotabuff.com/>.
7. OPENDOTA [Электронный ресурс] / OpenDota – Dota 2 Statistics. – Электрон. дан., 2019. – Режим доступа: https://www.opendota.com/.
8. Dota 2 – Таблицы лидеров [Электронный ресурс] / Dota 2 – Play for Free. – Элекрон. дан., 2019. – Режим доступа: <https://www.dota2.com/leaderboards/#europe>.
9. DOTA PLUS [Электронный ресурс] / Dota Plus – Электрон. дан., 2019. – Режим доступа: <https://www.dota2.com/plus>.
10. DOTAPICKER [Электронный ресурс] / Dota 2 hero pick web tools – Электрон. дан., 2019. – Режим доступа: <http://dotapicker.com/>.
11. Conley, K. How Does We Saw Me? A Recommendation Engine for Picking Heroes in Dota 2 / K. Conley, D. Perry. – Stanford University, 2013. – Режим доступа: <http://cs229.stanford.edu/proj2013/PerryConley-HowDoesHeSawMeARecommendationEngineForPickingHeroesInDota2.pdf>.
12. Johansson, F. Result Prediction by Mining Replays in Dota 2 / F. Johansson, J. Wikstrom. – Blekinge Institute of Technology, 2015. – Режим доступа: <https://www.diva-portal.org/smash/get/diva2:829556/FULLTEXT01.pdf>.
13. Kalyanaraman, K. To win or not to win? A prediction model to determine the outcome of a DotA2 match / K. Kalyanaraman. – University of California, San Diego, 2014. – Режим доступа: <https://pdfs.semanticscholar.org/7745/27ade8b86447c788a0d2b1618712c400e340.pdf>.
14. Kinkade, N. DOTA 2 Win Prediction / N. Kinkade, K. Lim. – University of California, San Diego, 2015. – Режим доступа: <http://jmcauley.ucsd.edu/cse255/projects/fa15/018.pdf>.
15. Song, K. Predicting the winning side of DotA2 / K. Song, T. Zhang, C. Ma. – Stanford University, 2015. – Режим доступа: <http://cs229.stanford.edu/proj2015/249_report.pdf>.
16. Уроки участия в хакатоне “Dota Science” в рамках “Data Fest 2” [Электронный ресурс] / Хабр – крупнейший в Европе ресурс для IT-специалистов, издаваемый компанией “TM” – Электрон. дан., 2016. – Режим доступа: <https://habr.com/ru/post/278785/>.
17. Сорокин, А.С. Построение скоринговых карт с использованием модели логистической регрессии / А.С. Сорокин // НАУКОВЕДЕНИЕ. – 2014. – № 2. – Режим доступа: <http://naukovedenie.ru/PDF/180EVN214.pdf>.
18. Разработка скоринговых моделей. Часть 1 [Презентация]. – Рязань.: BaseGroup Labs, 2016. – 67 с.
19. Пищулин, А. Очки для заемщика / А. Пищулин // Коммерсантъ Деньги. – 2008. – № 2. – С. 40. – Режим доступа: https://www.kommersant.ru/doc/842988.
20. Кредитный скоринг. Не все так страшно [Электронный ресурс] / Факторинг для профессионалов. Новости. Обзоры. Факторинговые компании. – Электрон. дан., 2008. – Режим доступа: <http://factoringpro.ru/index.php/credit-scoring-statya/408-skoring-statya-kredi/>.
21. Паклин, Н.Б. Бизнес-аналитика: от данных к знаниям (+CD): Учебное пособие. 2-е изд., испр. / Н.Б. Паклин, В.И. Орешков. – СПб.: Питер, 2013. – 704 с.
22. Качество данных [Электронный ресурс] / Качество данных | BaseGroup Labs. – Электрон. дан., 2019. – Режим доступа: <https://basegroup.ru/deductor/function/algorithm/data-quality>.
23. Репрезентативность (Represent) [Электронный ресурс] / Репрезентативность | Loginom Wiki. – Электрон. дан., 2019. – Режим доступа: <https://wiki.loginom.ru/articles/represent.html>.
24. Разработка скоринговых моделей. Часть 2 [Презентация]. – Рязань.: BaseGroup Labs, 2016 г. – 39 с.
25. Коэффициент WoE (Coefficient WOE) [Электронный ресурс] / Коэффициент WoE | Loginom Wiki. – Электрон. дан., 2019. – Режим доступа: https://wiki.loginom.ru/articles/coefficient-woe.html.
26. Коэффициент IV (Coefficient IV) [Электронный ресурс] / Коэффициент IV | Loginom Wiki. – Электрон. дан., 2019. – Режим доступа: https://wiki.loginom.ru/articles/coefficient-iv.html.
27. Гасанов, О.С. Скоринг при управлении кредитными рисками / О.С. Гасанов, Я.Р. Таранов // НАУКОВЕДЕНИЕ – 2016. – Т. 8, № 4. – Режим доступа: <http://naukovedenie.ru/PDF/37EVN416.pdf>.
28. Скоринг (scoring) [Электронный ресурс] / Банки.ру – финансовый супермаркет. Вклады, кредиты, ипотека, страховые и инвестиционные продукты. – Элекрон. дан., 2019. – Режим доступа: <https://www.banki.ru/wikibank/skoring/>.
29. Логистическая регрессия (Logistic Regression) [Электронный ресурс] / Логистическая регрессия | Loginom Wiki. – Электрон. дан., 2019. – Режим доступа: https://wiki.loginom.ru/articles/logistic-regression.html.
30. Сэмплинг (Sampling) [Электронный ресурс] / Сэмплинг | Loginom Wiki. – Электрон. дан., 2019. – Режим доступа: https://wiki.loginom.ru/articles/sampling.html.
31. Михайлева, Е.Г. Методология и методы социологических исследований: Учеб. пособие для студентов высш. учеб. заведений / Е.Г. Михайлева. – Харьков: Изд-во НУА, 2016. – 286 с.
32. Логистическая регрессия и ROC-анализ – математический аппарат [Электронный ресурс] / BaseGroup Labs. – Электрон. дан., 2006. – Режим доступа: https://basegroup.ru/community/articles/logistic.
33. Сорокин, А.С. К вопросу валидации модели логистической регрессии в кредитном скоринге / А.С. Сорокин // НАУКОВЕДЕНИЕ. – 2014. – № 2. – Режим доступа: <http://naukovedenie.ru/PDF/173EVN214.pdf>.
34. Разработка скоринговых моделей. Часть 3 [Презентация]. – Рязань: BaseGroup Labs, 2016. – 102 с.
35. Сомасегар, С. Руководство Microsoft по проектированию архитектуры приложений, 2-е изд. / С. Сомасегар, Г. Скотт. – Пер. с англ. – США: Microsoft Press, 2009. – 529 с.
36. Информационный критерий Акаике (Akaike’s information criterion) [Электронный ресурс] / Информационный критерий Акаике | Loginom Wiki. – Электрон. дан., 2019. – Режим доступа: <https://wiki.loginom.ru/articles/aic.html>.
37. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы: Постановление Главного государственного санитарного врача РФ от 3 июня 2003 г. № 118. – М.: Министерство здравоохранения РФ, 2003. – 31 с.
38. ГОСТ 12.1.033-81. Система стандартов безопасности труда. Пожарная безопасность. Термины и определения (с Изменением № 1). – Введ. 01.07.1982. – М.: Изд-во стандартов, 2001. – 13 с.
39. СНиП 21-01-97. Пожарная безопасность зданий и сооружений (с Изменениями № 1, 2). Строительные нормы и правила. – Взамен СНиП 2.01.02-85; Введ. 01.01.1998. – М.: Минстрой России, 2002. – 17 c.
40. Фаулер, М. UML Основы: краткое руководство по стандартному языку объектного моделирования, 3-е изд. / М. Фаулер. – Пер. с англ. – СПб.: Символ-Плюс, 2004. – 192 с.
41. Лутц, М. Изучаем Python, 4-е изд. / М. Лутц. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
42. Requests: The User Guide [Электронный ресурс] / Requests: HTTP for HumansTM – Requests 2.21.0 documentation. – Электрон. дан., 2019. – Режим доступа: <http://docs.python-requests.org/en/master/>.
43. Nixon, R. Learning PHP, MySQL & JavaScript, 5th Edition. / R. Nixon. – USA: O’Reilly Media, 2018. – 832 p.

# ПРИЛОЖЕНИЕ А

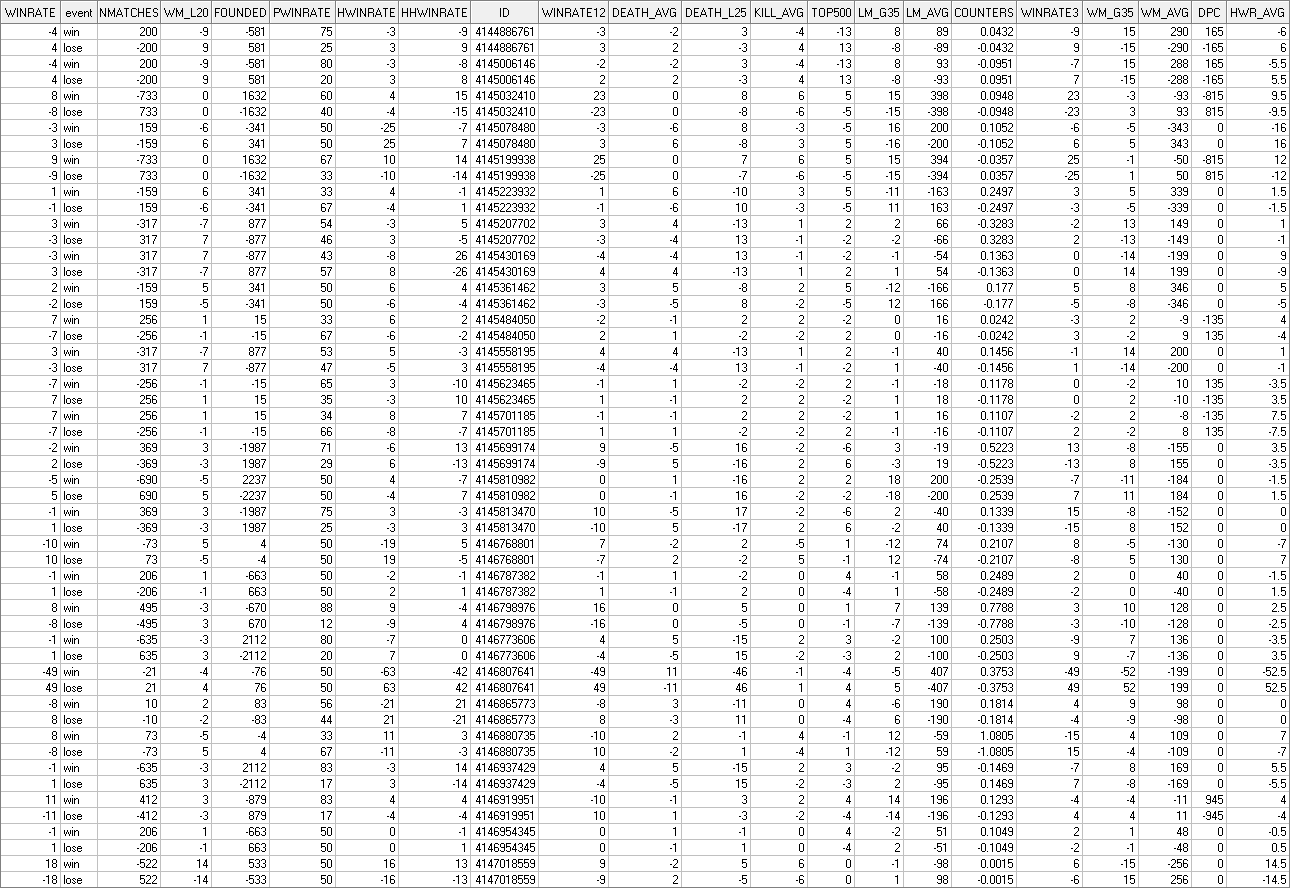


Рис. А.1. Фрагмент файла “SAMPLE.txt”