|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Системы обработки информации и управления»

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОМУ ПРОЕКТУ***

***НА ТЕМУ:***

***\_­­\_\_\_\_\_\_\_Решение задачи машинного обучения\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент группы ИУ5-61Б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Крюков Г. М.

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсового проекта **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_**Гапанюк Ю. Е.

(Подпись, дата) (И.О.Фамилия)

Консультант **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*Москва, 2020 г.*

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(И.О.Фамилия)

« \_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ г.

**ЗАДАНИЕ**

**на выполнение курсового проекта**

по дисциплине «Технологии машинного обучения»

Студент группы ИУ5-61Б

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Крюков Геннадий Максимович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Тема курсового проекта: “Бинарная классификация”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Направленность КП (учебный, исследовательский, практический, производственный, др.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Источник тематики (кафедра, предприятие, НИР) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

График выполнения проекта: 25% к \_\_\_ нед., 50% к \_\_\_ нед., 75% к \_\_ нед., 100% к \_\_\_ нед.

***Задание*** решение задачи машинного обучения на основе материалов дисциплины. Выполняется студентом единолично.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Оформление курсового проекта:***

Расчетно-пояснительная записка на \_\_\_42\_\_ листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «12» февраля 2020 г.

**Руководитель курсового проекта**  \_\_\_\_\_\_\_\_\_\_\_\_ Гапанюк Ю. Е.

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_** Крюков Г. М.

(Подпись, дата) И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

**Содержание**

[Задание 4](#_Toc10451934)

[Введение 5](#_Toc10451935)

[Основная часть 6](#_Toc10451936)

[Постановка задачи 6](#_Toc10451937)

[Решение поставленной задачи 6](#_Toc10451939)

[Заключение 41](#_Toc10451944)

[Список литературы 42](#_Toc10451945)

# **Задание**

В данной курсовой работе необходимо предпринять следующие шаги:

1. Поиск и выбор набора данных для построения моделей машинного обучения. На основе выбранного набора данных студент должен построить модели машинного обучения для решения или задачи классификации, или задачи регрессии.
2. Проведение разведочного анализа данных. Построение графиков, необходимых для понимания структуры данных. Анализ и заполнение пропусков в данных.
3. Выбор признаков, подходящих для построения моделей. Кодирование категориальных признаков Масштабирование данных. Формирование вспомогательных признаков, улучшающих качество моделей.
4. Проведение корреляционного анализа данных. Формирование промежуточных выводов о возможности построения моделей машинного обучения.

В зависимости от набора данных, порядок выполнения пунктов 2, 3, 4 может быть изменен.

1. Выбор метрик для последующей оценки качества моделей. Необходимо выбрать не менее трех метрик и обосновать выбор.
2. Выбор наиболее подходящих моделей для решения задачи классификации или регрессии. Необходимо использовать не менее пяти моделей, две из которых должны быть ансамблевыми.
3. Формирование обучающей и тестовой выборок на основе исходного набора данных.
4. Построение базового решения (baseline) для выбранных моделей без подбора гиперпараметров. Производится обучение моделей на основе обучающей выборки и оценка качества моделей на основе тестовой выборки.
5. Подбор гиперпараметров для выбранных моделей. Рекомендуется использовать методы кросс-валидации. В зависимости от используемой библиотеки можно применять функцию GridSearchCV, использовать перебор параметров в цикле, или использовать другие методы.
6. Повторение пункта 8 для найденных оптимальных значений гиперпараметров. Сравнение качества полученных моделей с качеством baseline-моделей.
7. Формирование выводов о качестве построенных моделей на основе выбранных метрик. Результаты сравнения качества рекомендуется отобразить в виде графиков и сделать выводы в форме текстового описания. Рекомендуется постройение графиков обучения и валидации, влияния значений гиперпарметров на качество моделей и т.д.

# **Введение**

Курсовая работа – самостоятельная часть учебной дисциплины «Технологии машинного обучения» – учебная и практическая исследовательская студенческая работа, направленная на решение комплексной задачи машинного обучения. Результатом курсовой работы является отчет, содержащий описания моделей, тексты программ и результаты экспериментов.

Курсовая работа опирается на знания, умения и владения, полученные студентом в рамках лекций и лабораторных работ по дисциплине.

В рамках данной курсовой работы необходимо применить навыки, полученные в течение курса «Технологии машинного обучения», и обосновать полученные результаты.

# **Основная** **часть**

# **Постановка задачи**

В данной курсовой работе ставится задача определения принадлежности звезды к классу пульсаров по различным параметрам с помощью методов машинного обучения: "Logistic Regression", "Support vector machine", "Decision tree", "Gradient boosting", "Random forest". С помощью различных метрик выбор метода, который наиболее эффективно и качественно определяет значение целевого признака.

# **Решение поставленной задачи**

1. **Поиск и выбор набора данных для построения моделей машинного обучения. На основе выбранного набора данных студент должен построить модели машинного обучения для решения или задачи классификации, или задачи регрессии**

# Описание выбранного датасета

Выбранный набор данных описывает статистику игр лучших игроков в дисциплине League of Legends.

**League of Legends** — ролевая видеоигра с элементами стратегии в реальном времени (MOBA), разработанная и выпущенная компанией Riot Games 27 октября 2009 года. Игра распространяется по модели free-to-play. Ежемесячная аудитория игры составляет 100 млн игроков по всему миру (сентябрь 2016).

Подобно игровым видам спорта, игра в League of Legends происходит в виде отдельных партий (сессий). Чтобы сыграть игру в определённом режиме, необходимо подать заявку и дождаться автоматического подбора игроков для следующей партии. Подбор осуществляется с учетом рейтинга игрока в соответствующем режиме. Поиск игры длится около минуты, сама же игровая сессия обычно длится 16-65 минут. Каждая игра начинается «с чистого листа», то есть все игроки имеют одинаковый наименьший начальный уровень, стартовое количество ресурсов.

Сейчас турниры League of Legends являются одними из самых массовых и зрелищных.

Трансляции финала мирового чемпионата Лиги Легенд третьего сезона (2013 год) посмотрело 32 миллиона зрителей, что сделало его самым популярным (к тому времени) киберспортивным событием в истории.

В 2013 году США признало профессиональных игроков League of Legends в качестве спортсменов, с возможностью выдачи им виз P-1.

Информация об атрибутах:

Каждая игра описывается 49 непрерывными переменными и одной переменной класса. Первые 2 являются «общими» параметрами. 24 – описания для одной из команд, другие 24 – описание для другой (причем один из них является целевым признаком).

Всего примеров 26 854.

* 13 454 – победа синей команды.
* 13 450 – победа красной команды.

В рассматриваемом примере будем решать задачу классификации:

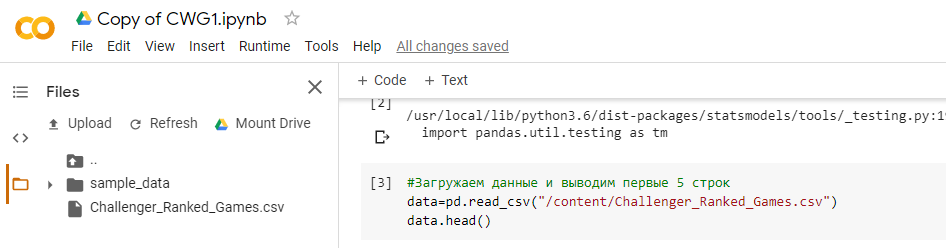
* Для решения **задачи классификации** в качестве целевого признака будем использовать "blueWins" (Класс). Поскольку признак содержит только значения 0 и 1, то это задача бинарной классификации.

Импорт библиотек



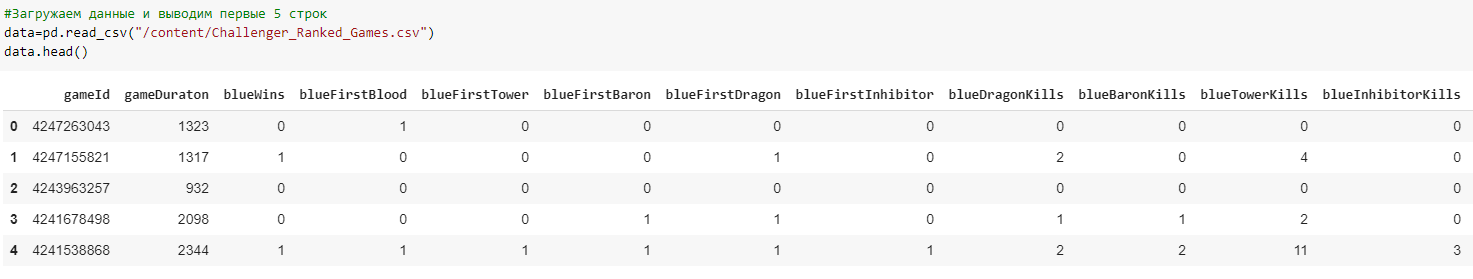
Загрузка данных.

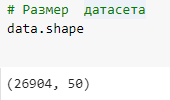
Загрузим файл датасета c помощью библиотеки Pandas.



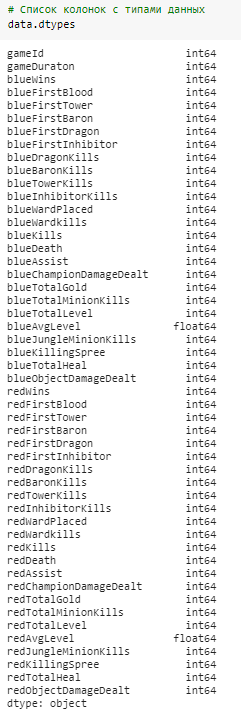
1. **Проведение разведочного анализа данных. Построение графиков, необходимых для понимания структуры данных. Анализ и заполнение пропусков в данных.**

Основные характеристики датасета

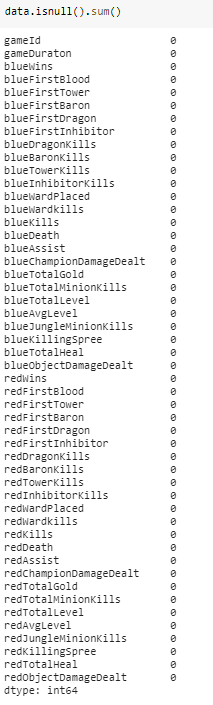
****

****

****

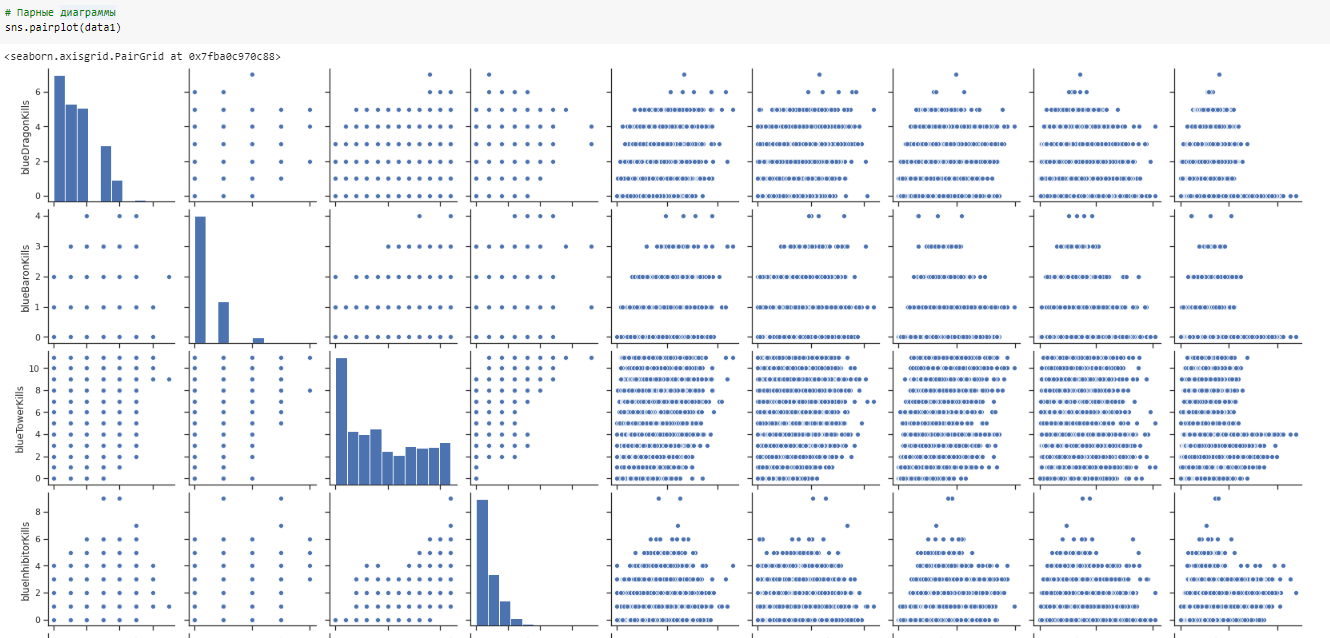
****

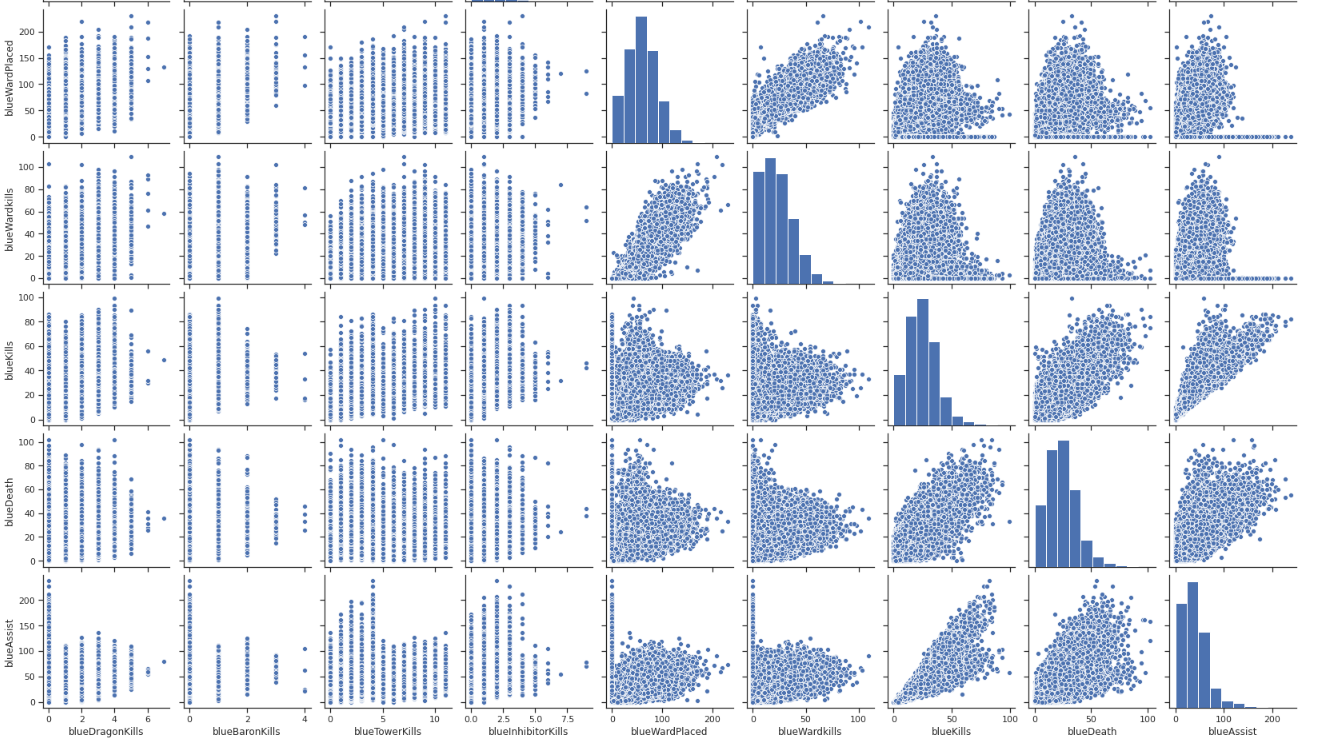
Посмотрим заполненность датасета. Возможно есть пропуски.

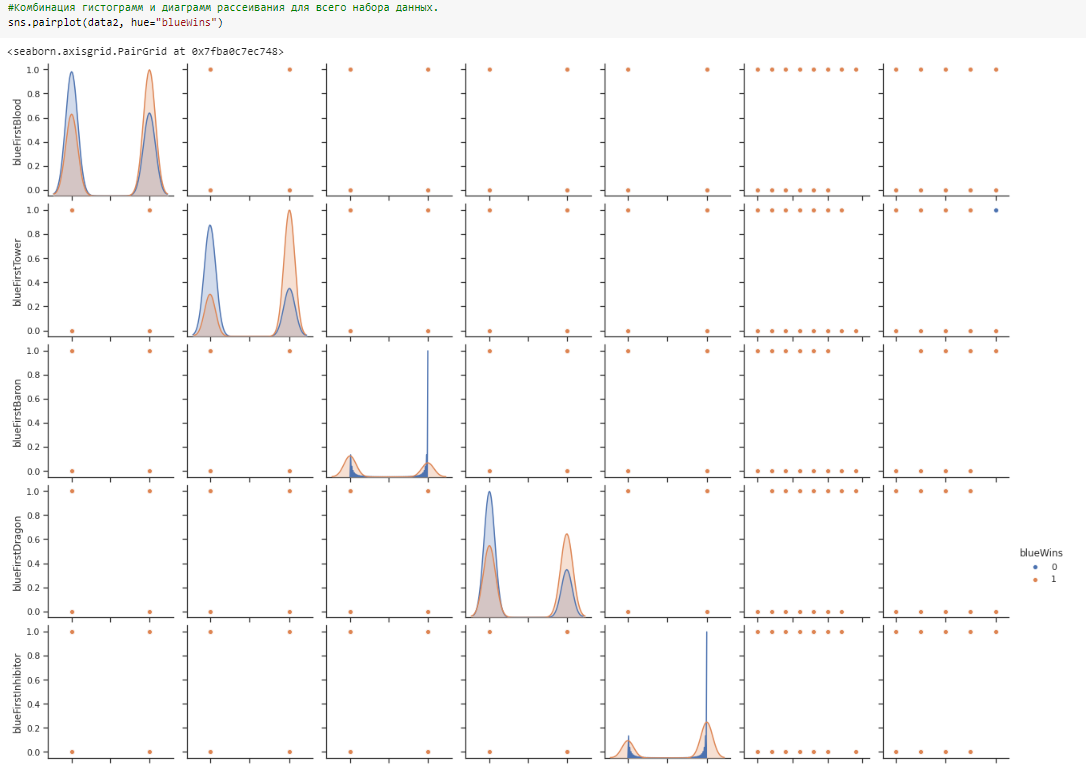


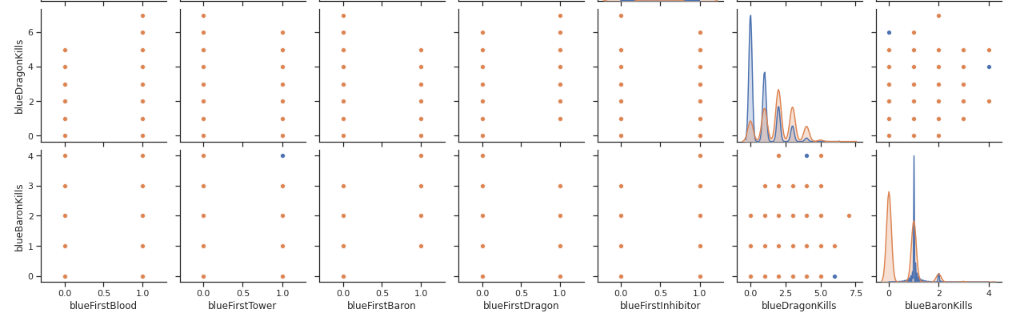
Пропуски данных отсутствуют.

Построим некоторые графики для понимания структуры данных.



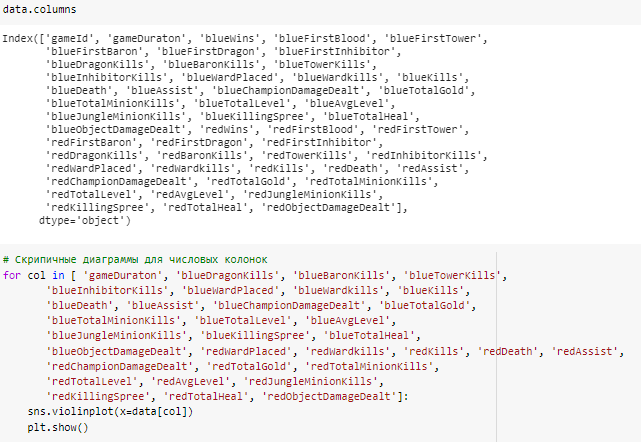


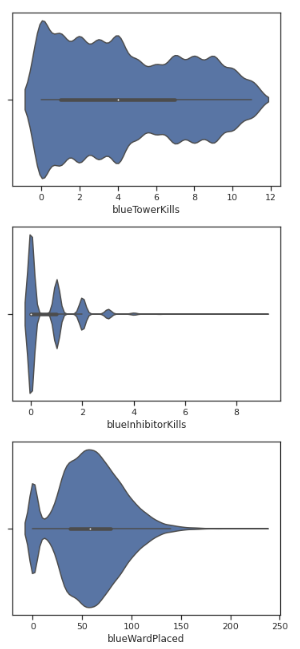
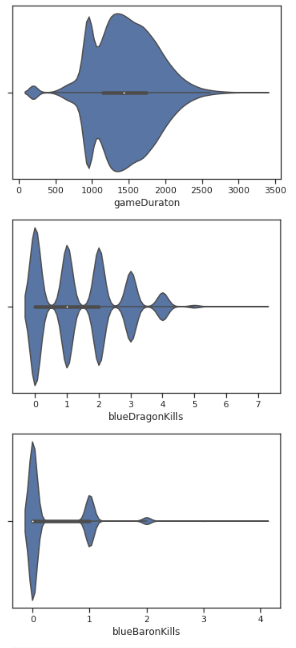


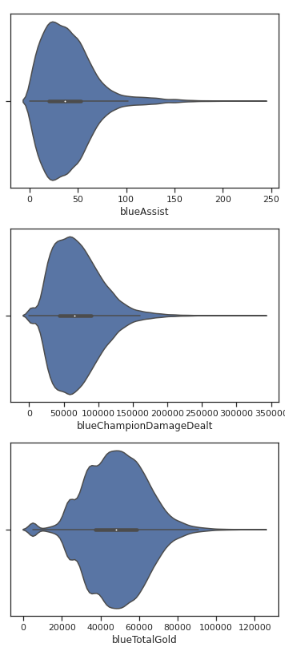
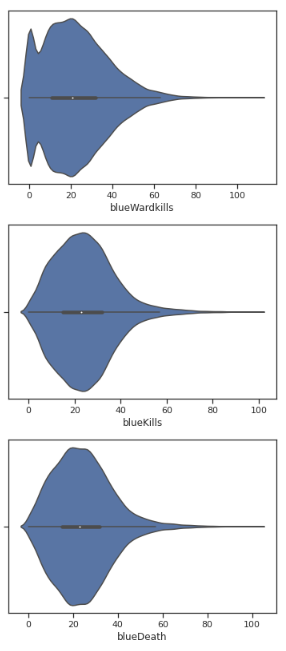


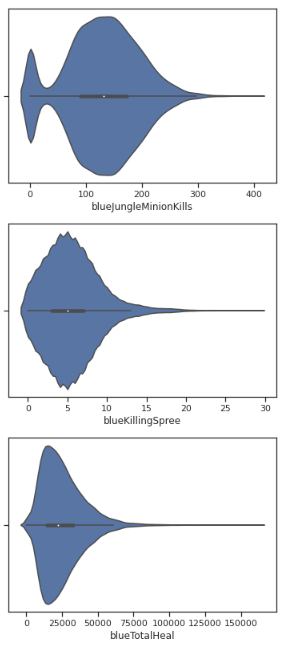
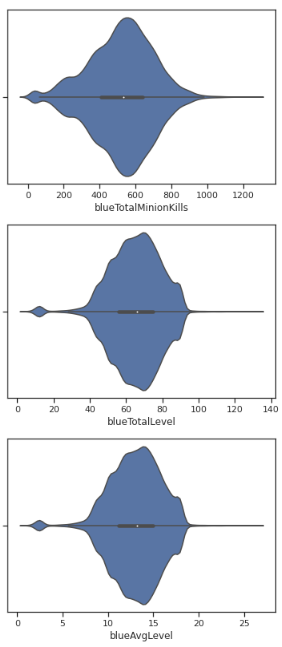


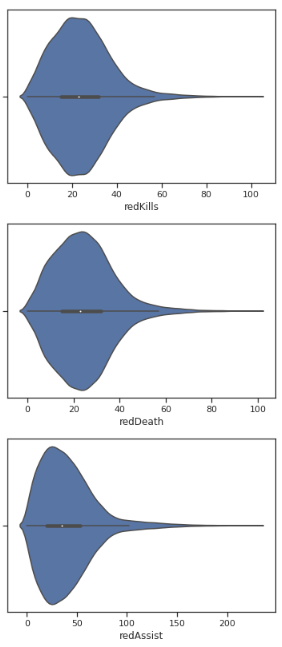
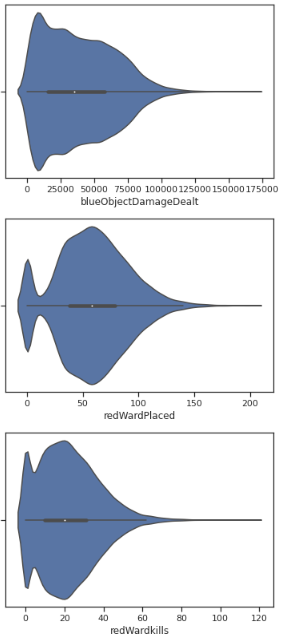
Дисбаланс классов практически отсутствует.

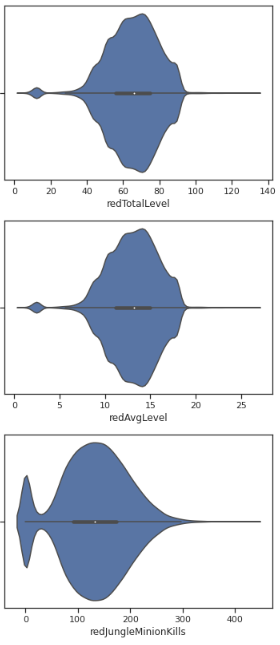
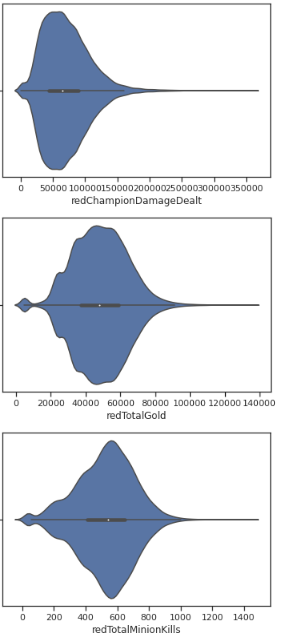


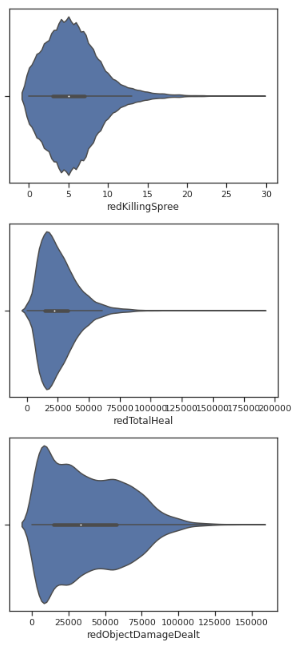












1. **Выбор признаков, подходящих для построения моделей. Кодирование категориальных признаков. Масштабирование данных. Формирование вспомогательных признаков, улучшающих качество моделей.**

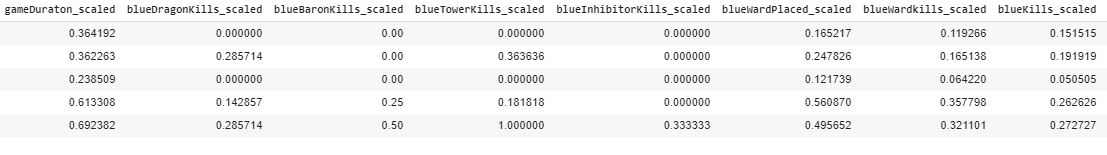
Для построения моделей будем использовать все признаки.

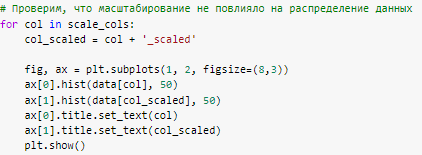
Категориальные признаки отсутствуют, их кодирования не требуется. Исключением является признак target\_class, но в представленном датасете он уже закодирован на основе подхода LabelEncoding.

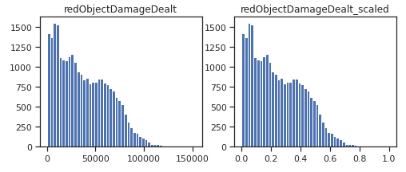
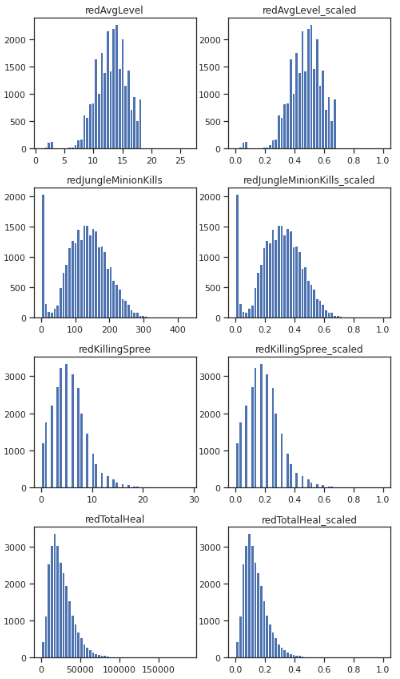
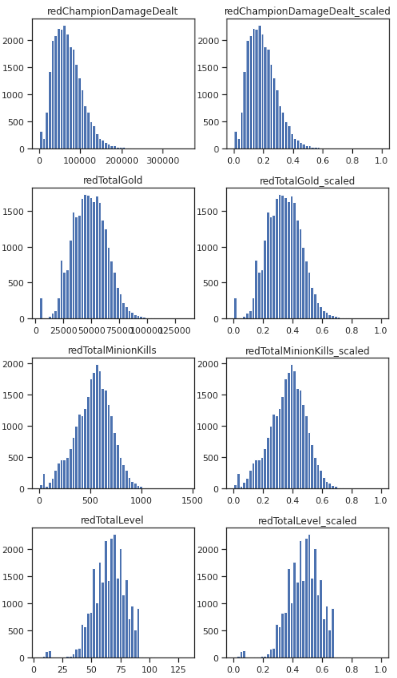
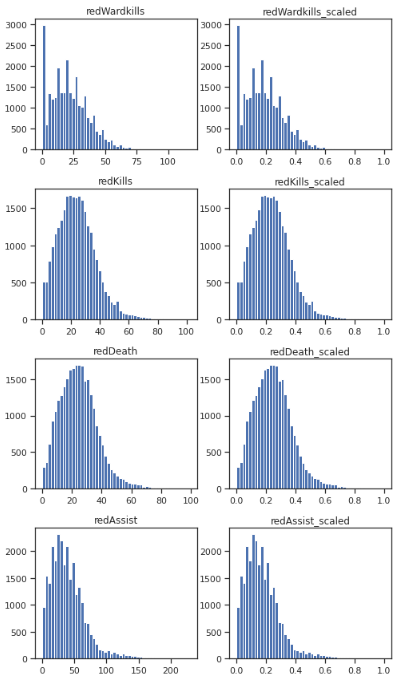
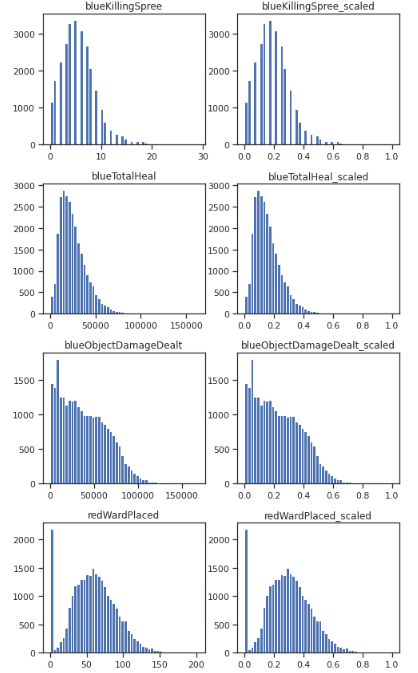
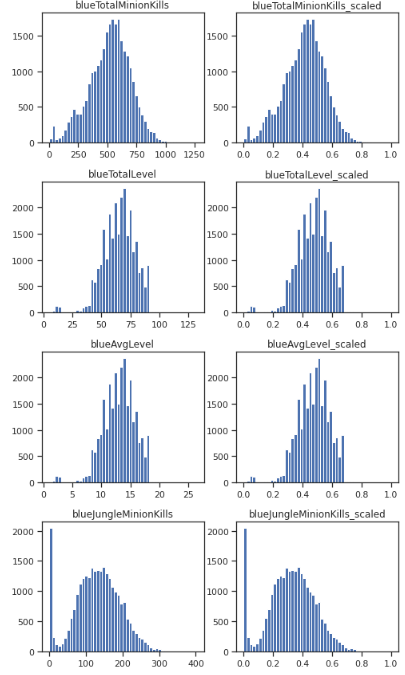
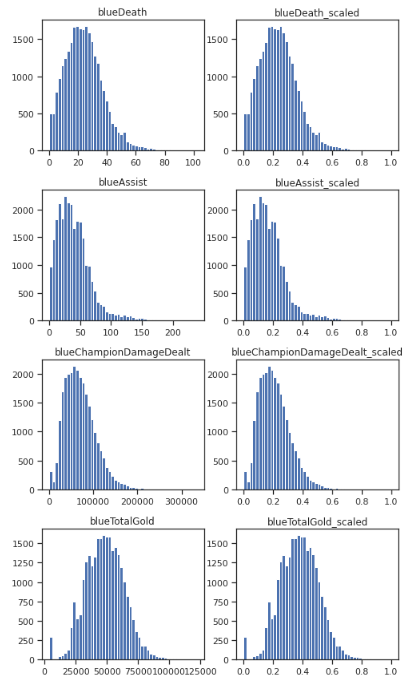
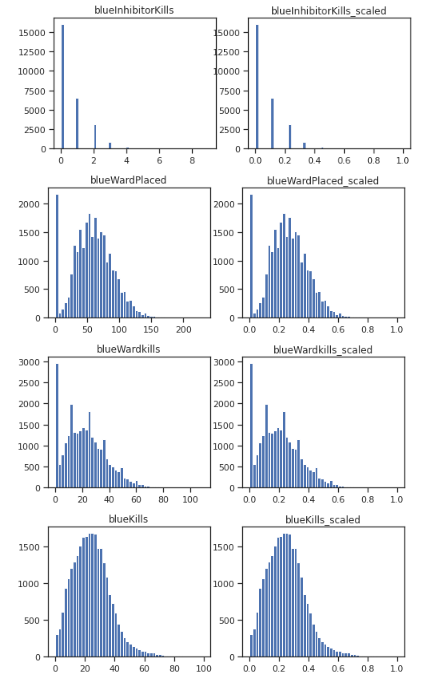
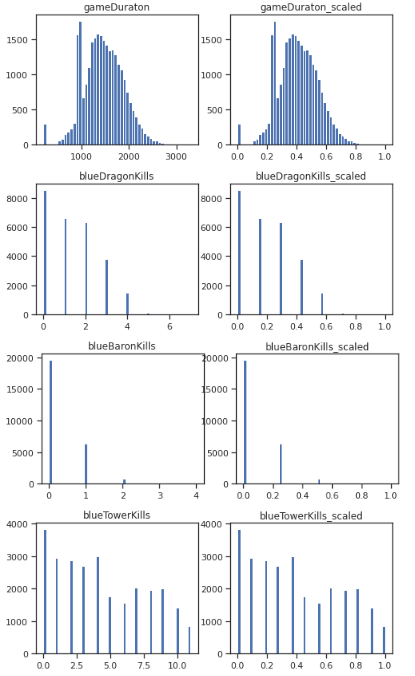
Вспомогательные признаки для улучшения качества моделей строить не будем.

Выполним масштабирование данных.

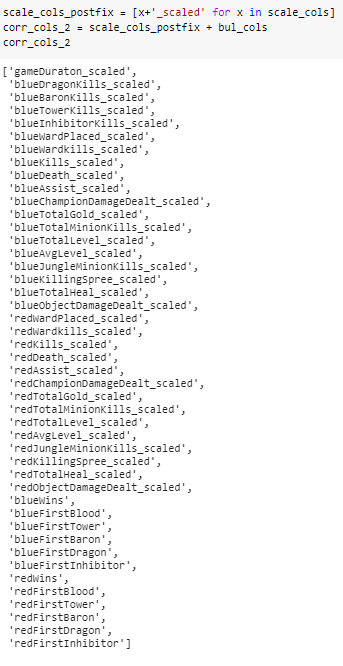
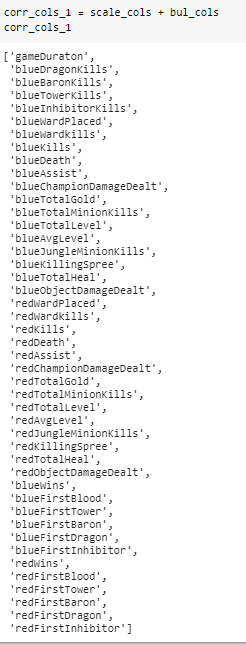
****

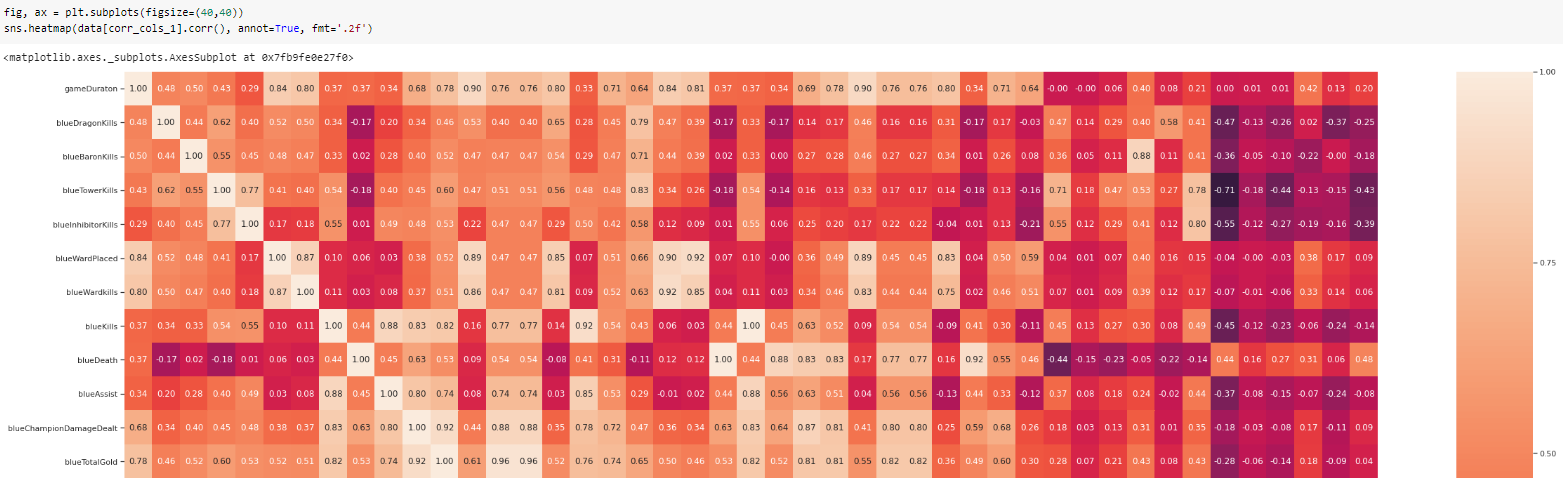
****

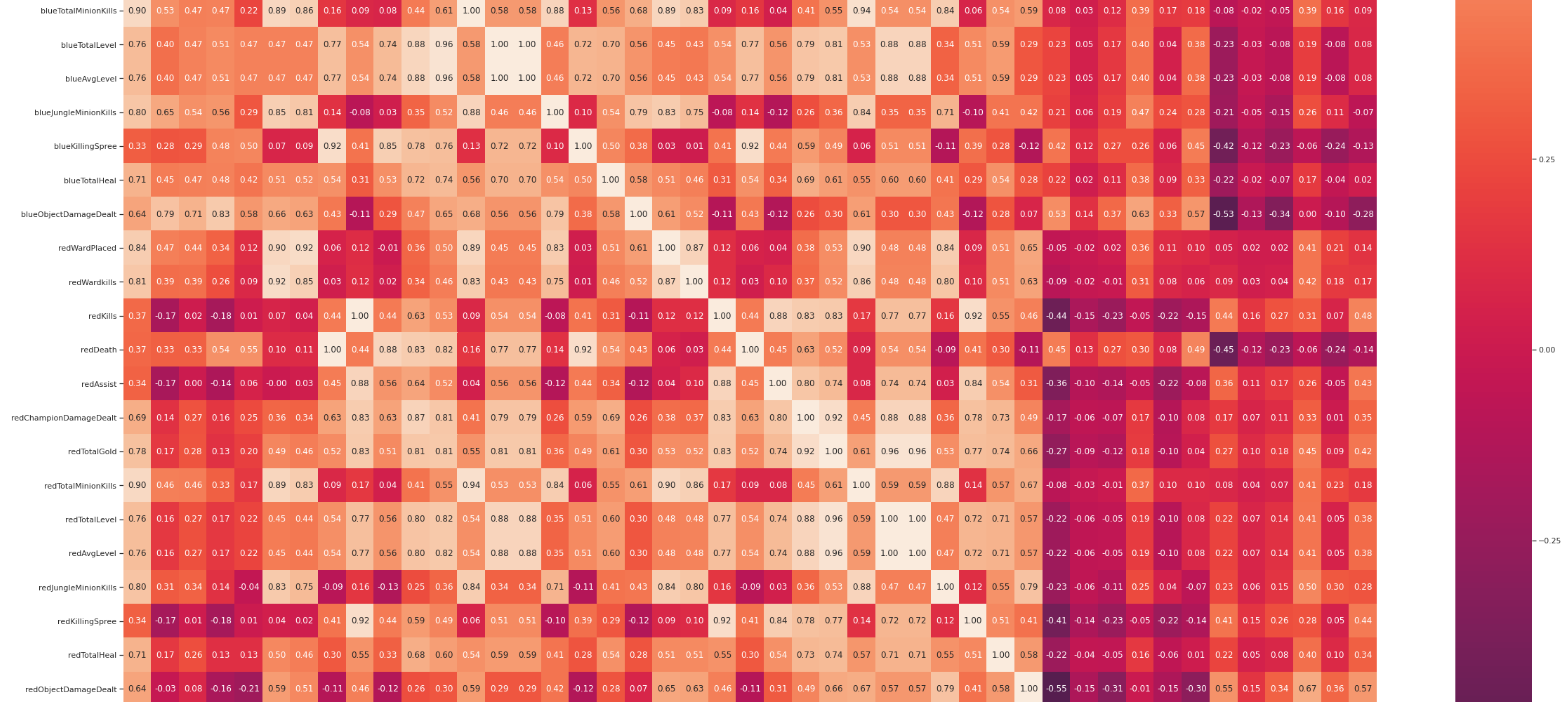
****

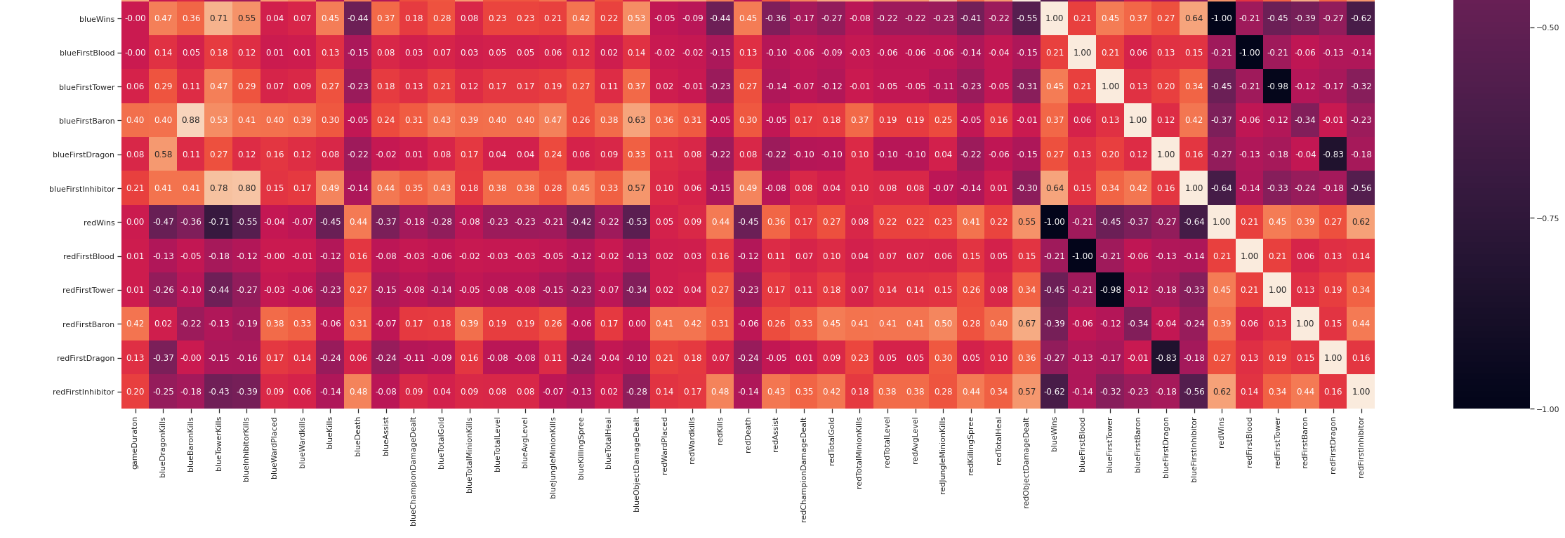


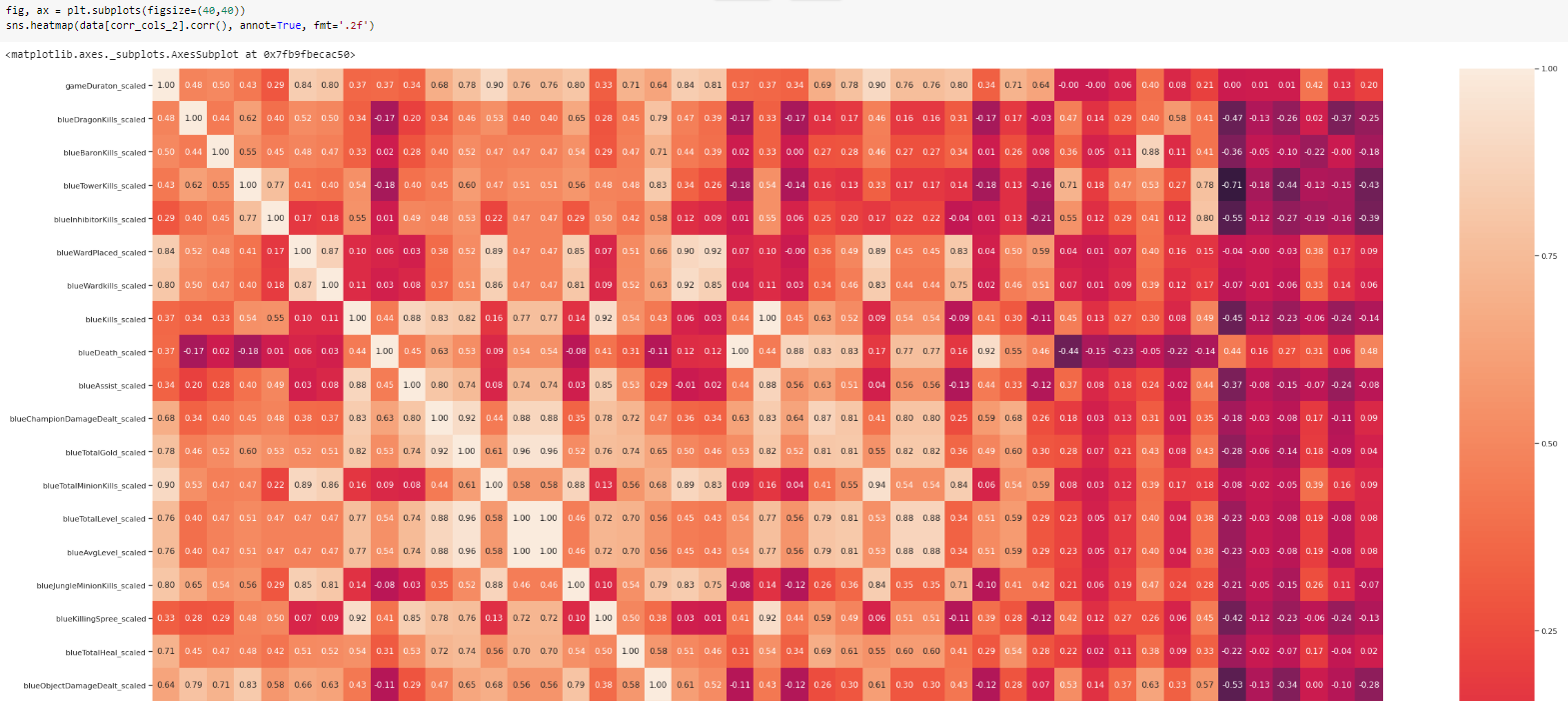
1. **Проведение корреляционного анализа данных. Формирование промежуточных выводов о возможности построения моделей машинного обучения.**

****

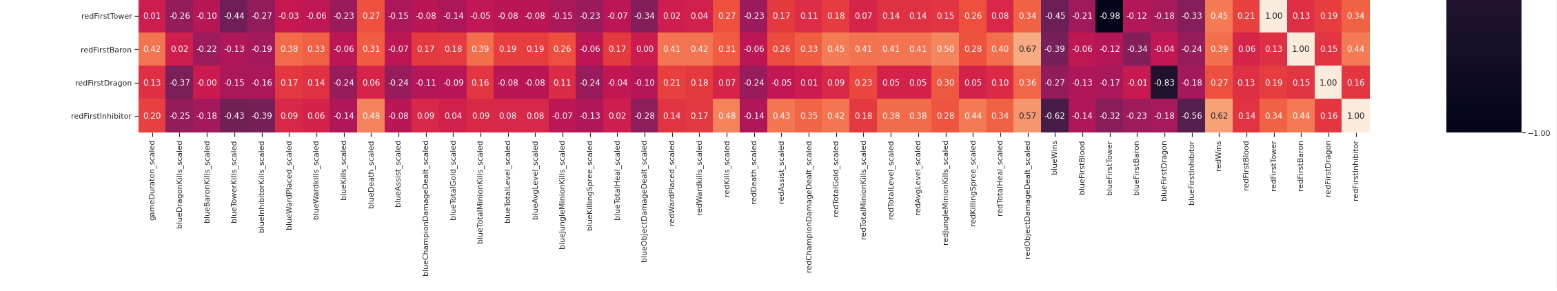
****

****

****

****

****

****

На основе корреляционной матрицы можно сделать следующие выводы:

1. Корреляционные матрицы для исходных и масштабированных данных совпадают.
2. Целевой признак классификации "blueWins" наиболее сильно коррелирует с признаками:

«blueTowerKills» (Синяя команда уничтожила вражеских строений) - 0.71

«blueInhibitorKills» (Синяя команда уничтожила кристаллов усиления) - 0.55

«blueKills» (Синяя команда уничтожила вражеских чемпионов) - 0.45

«blueAssist» (Игрок синей команды помог уничтожить вражеского чемпиона) - 0.37

«blueKillingSpree» (Игрок синей команды совершил массовое уничтожение) - 0.42

«blueObjectDamageDealt» (Синяя команда нанесла урона игровым объектам) - 0.53

«redDeath» (Игроки синей команды были уничтожены) - 0.45

«blueFirstTower» (Синяя команда первой сломала башню) - 0.45

«blueFirstBaron» (Синяя команда получила бонус Барона Нашора первой) - 0.37

«blueFirstInhibitor» (Синяя команда уничтожила кристалл усиления первой) - 0.64

Эти признаки обязательно следует оставить в модели классификации.

Достаточно большие по модулю значения коэффициентов корреляции свидетельствуют о значимой корреляции между исходными признаками и целевым признаком. На основании корреляционной матрицы можно сделать вывод о том, что данные позволяют построить модель машинного обучения.

1. **Выбор метрик для последующей оценки качества моделей.**

#### В качестве метрик для решения задачи классификации будем использовать:

#### Метрика precision:

#### 

#### Доля верно предсказанных классификатором положительных объектов, из всех объектов, которые классификатор верно или неверно определил как положительные.

#### Используется функция *precision\_score*.

#### Метрика recall (полнота):

#### 

#### Доля верно предсказанных классификатором положительных объектов, из всех действительно положительных объектов.

#### Используется функция *recall\_score*.

#### Метрика F1-мера

#### Для того, чтобы объединить precision и recall в единую метрику используется Fβ-мера, которая вычисляется как среднее гармоническое от precision и recall:

#### 

#### На практике чаще всего используют вариант F1-меры (которую часто называют F-мерой) при β=1:

#### 

#### Для вычисления используется функция *f1\_score*.

#### Метрика ROC AUC

#### Используется для оценки качества бинарной классификации. Основана на вычислении следующих характеристик:

#### 

#### True Positive Rate, откладывается по оси ординат. Совпадает с recall.

#### False Positive Rate, откладывается по оси абсцисс. Показывает какую долю из объектов отрицательного класса алгоритм предсказал неверно.

#### Чем сильнее отклоняется кривая от верхнего левого угла графика, тем хуже качество классификации.

#### В качестве количественной метрики используется площадь под кривой - ROC AUC (Area Under the Receiver Operating Characteristic Curve). Чем ниже проходит кривая тем меньше ее площадь и тем хуже качество классификатора.

#### Для получения ROC AUC используется функция *roc\_auc\_score*.

#### Сохранение и визуализация метрик

Разработаем класс, который позволит сохранять метрики качества построенных моделей и реализует визуализацию метрик качества.



1. **Выбор наиболее подходящих моделей для решения задачи классификации или регрессии.**

Для задачи классификации будем использовать следующие модели:

* Логистическая регрессия

Метод, используемый для решения задачи бинарной классификации.

Метод выдает вероятность принадлежности объекта к нулевому/единичному классам.

Используется класс *LogisticRegression*.

* Машина опорных векторов

Основная идея метода — перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве. Две параллельных гиперплоскости строятся по обеим сторонам гиперплоскости, разделяющей классы. Разделяющей гиперплоскостью будет гиперплоскость, максимизирующая расстояние до двух параллельных гиперплоскостей. Алгоритм работает в предположении, что чем больше разница или расстояние между этими параллельными гиперплоскостями, тем меньше будет средняя ошибка классификатора.

Для решения задачи классификации используем класс:

*SVC* - основной классификатор на основе SVM. Поддерживает различные ядра.

* Решающее дерево

Для текущего выбранного признака (колонки) из N признаков построить все варианты ветвления по значениям (для категориальных признаков) или по диапазонам значений (для числовых признаков).

Если подвыборке соответствует единственное значение целевого признака, то в дерево добавляется терминальный лист, который соответствует предсказанному значению.

Если в подвыборке больше одного значения целевого признака, то предыдущие пункты выполняются рекурсивно для подвыборки.

Для решения задачи классификации используется класс *DecisionTreeClassifier*.

* Случайный лес (ансамблевая)

Случайный лес можно рассматривать как алгоритмом бэггинга над решающими деревьями.

Но при этом каждое решающее дерево строится на случайно выбранном подмножестве признаков. Эта особенность называется "feature bagging" и основана на методе случайных подпространств.

Случайный лес для задачи классификации реализуется в scikit-learn с помощью класса *RandomForestClassifier*.

Задание параметра n\_jobs=-1 распараллеливает алгоритм на максимально возможное количество процессоров.

* Градиентный бустинг (ансамблевая)

В отличие от методов бэггинга и случайного леса, которые ориентированы прежде всего на минимизизацию дисперсии (Variance), методы бустинга ориентированы прежде всего на минимизацию смещения (Bias) и, отчасти, на минимизизацию дисперсии.

Исторически первым полноценным алгоритмом бустинга считается алгоритм AdaBoost.

AdaBoost реализуется в scikit-learn с помощью класса *AdaBoostClassifier* для задач классификации.

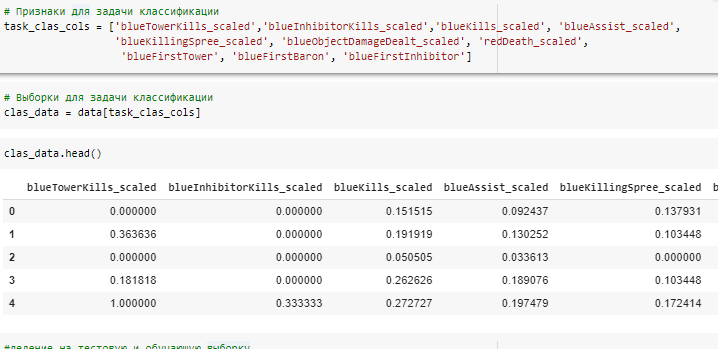
1. **Формирование обучающей и тестовой выборок на основе исходного набора данных.**

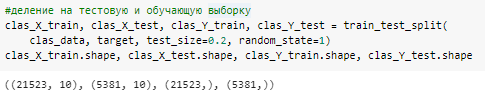
Имеем масштабированные данные:



****

На основе масштабированных данных выделим обучающую и тестовую выборки:

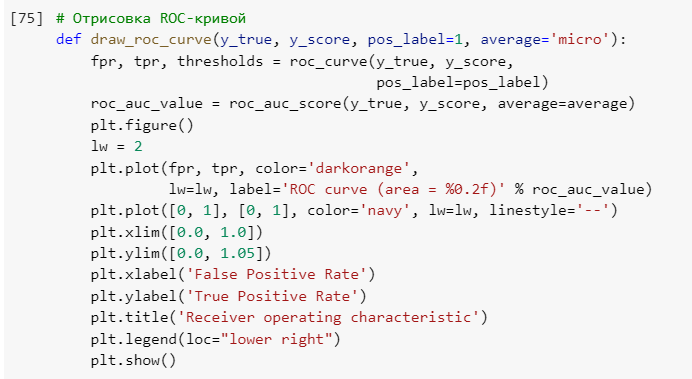


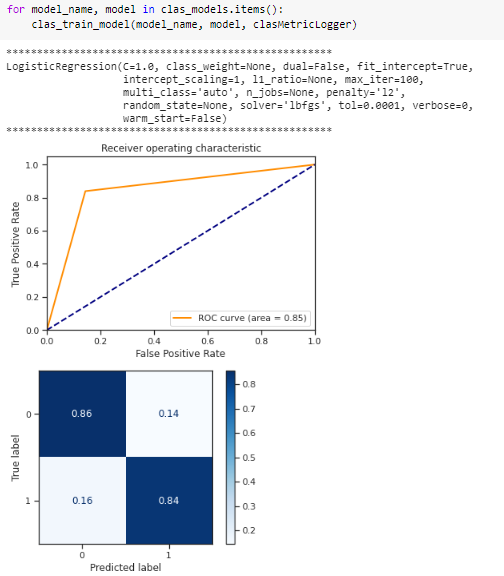


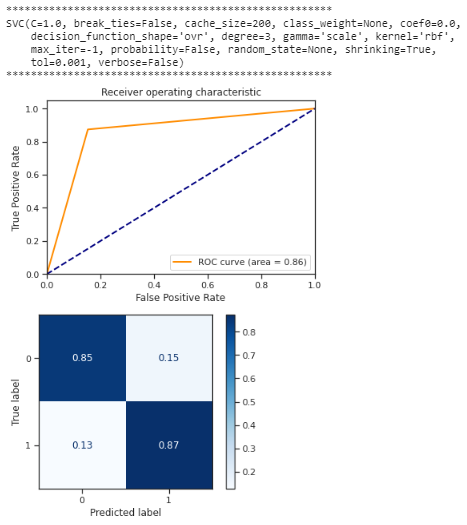
1. **Построение базового решения (baseline) для выбранных моделей без подбора гиперпараметров. Производится обучение моделей на основе обучающей выборки и оценка качества моделей на основе тестовой выборки.**

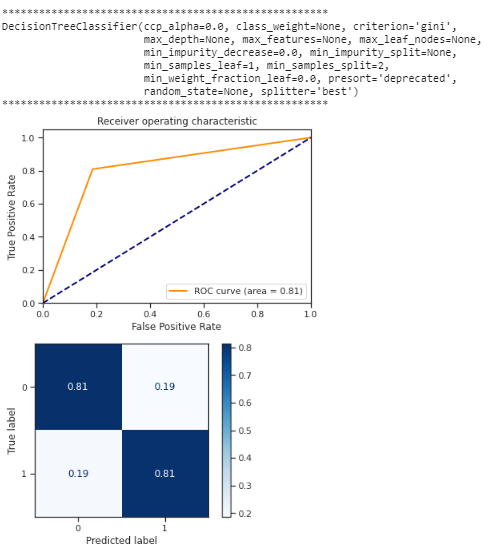
****

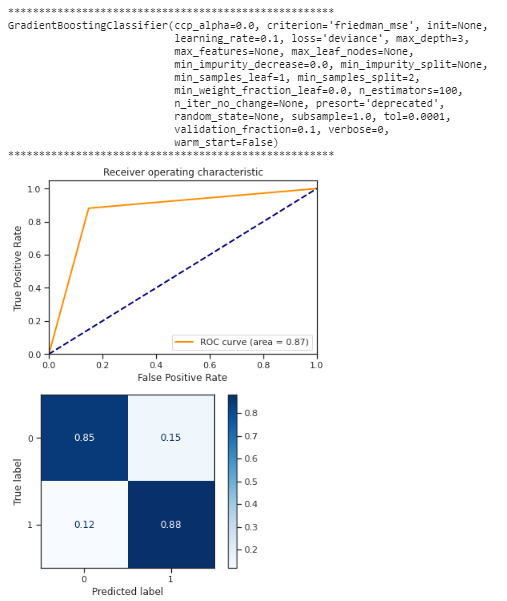
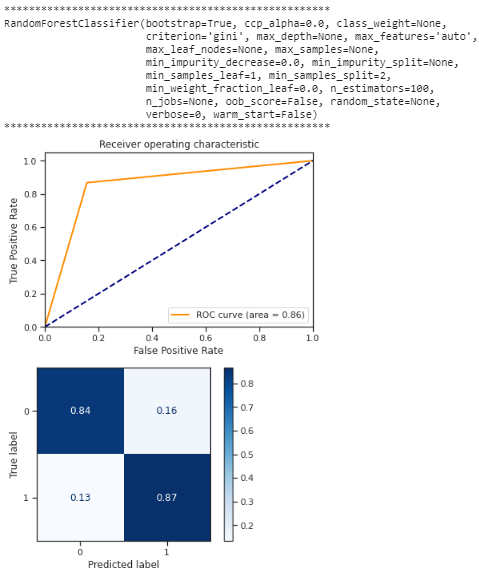




****

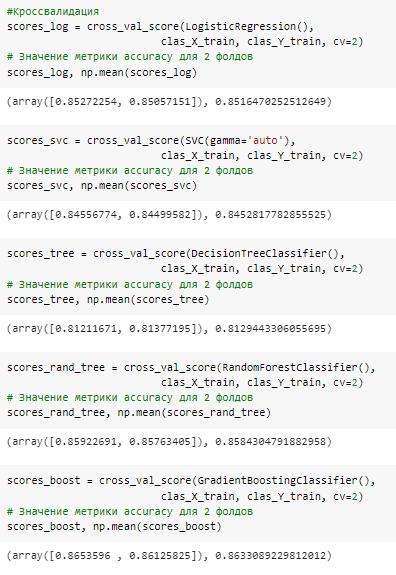
****

****

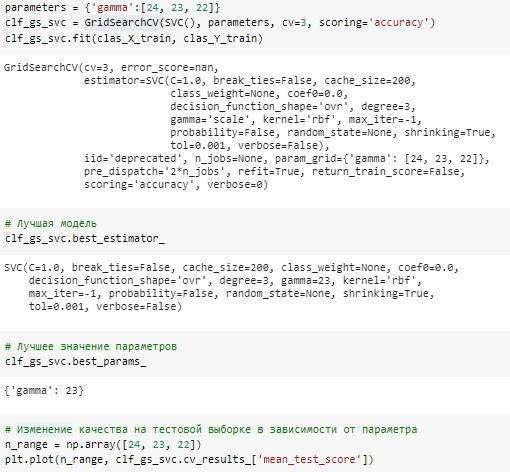
****

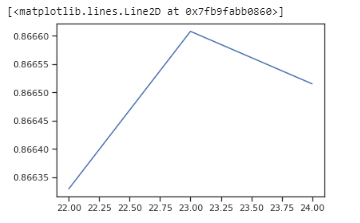
1. **Подбор гиперпараметров для выбранных моделей. Рекомендуется использовать методы кросс-валидации. В зависимости от используемой библиотеки можно применять функцию GridSearchCV, использовать перебор параметров в цикле, или использовать другие методы.**

Кросс-валидация

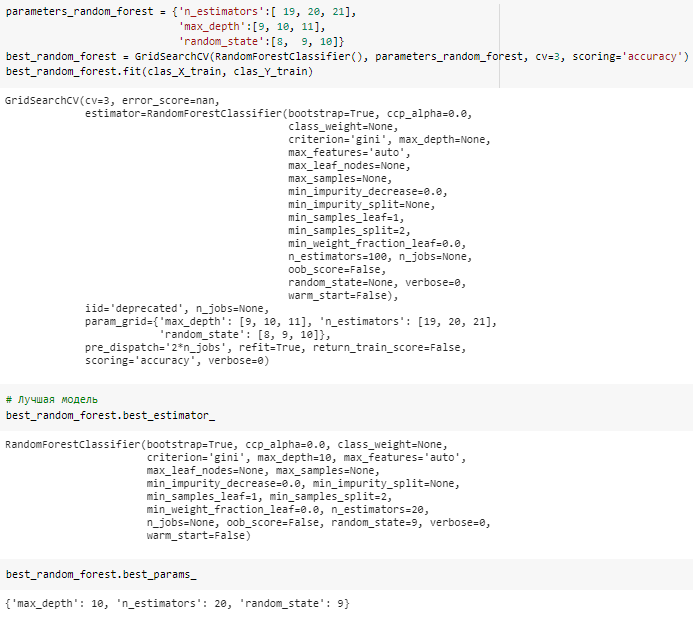
****

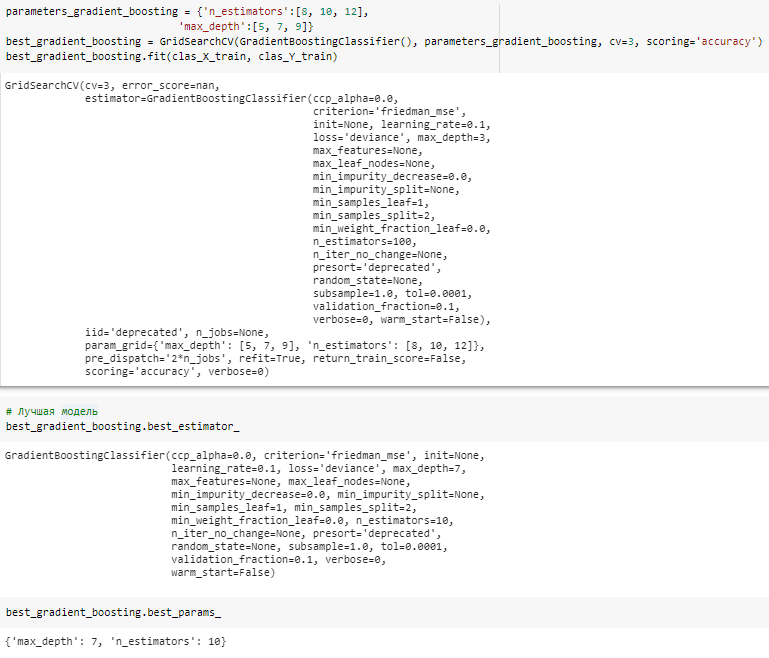
Подбор гиперпараметров:



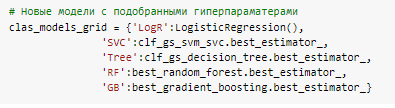


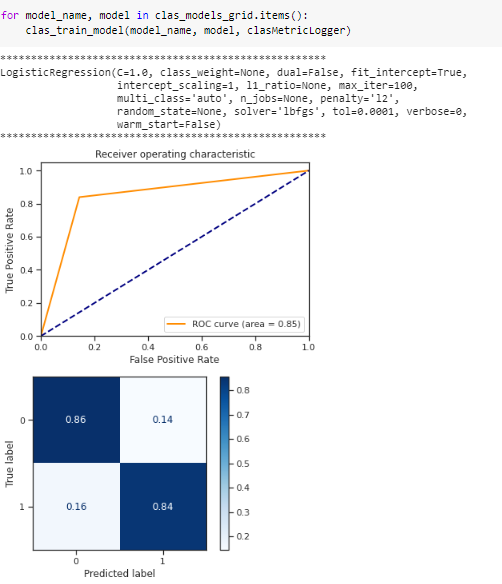


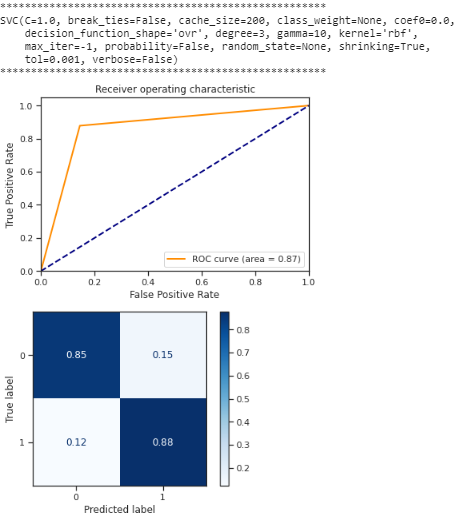


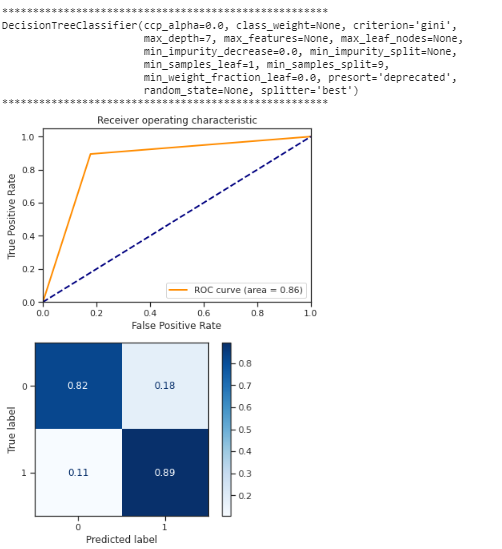


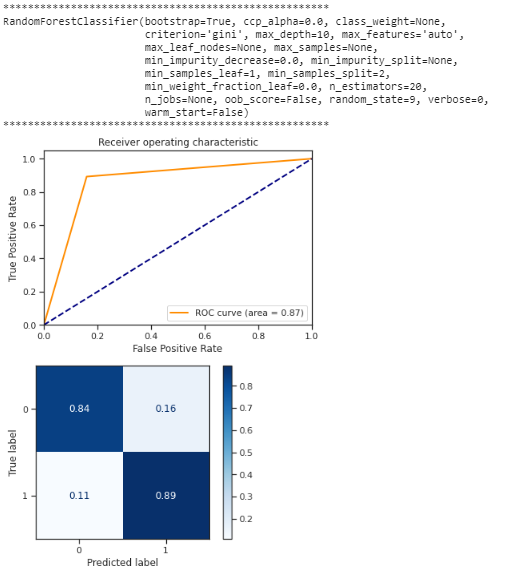
1. **Повторение пункта 8 для найденных оптимальных значений гиперпараметров. Сравнение качества полученных моделей с качеством baseline-моделей.**

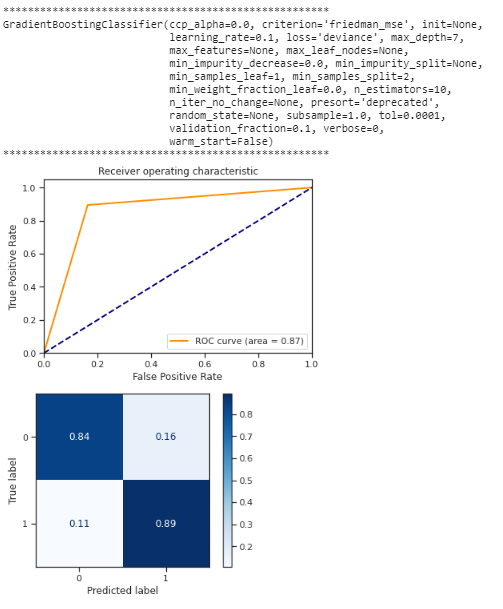






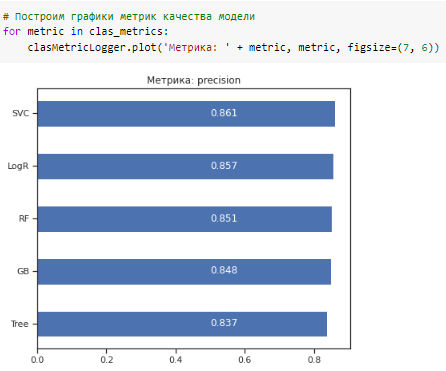




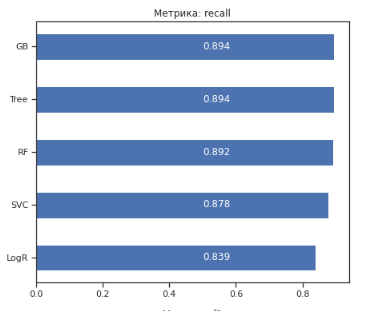


1. **Формирование выводов о качестве построенных моделей на основе выбранных метрик. Результаты сравнения качества рекомендуется отобразить в виде графиков и сделать выводы в форме текстового описания.**

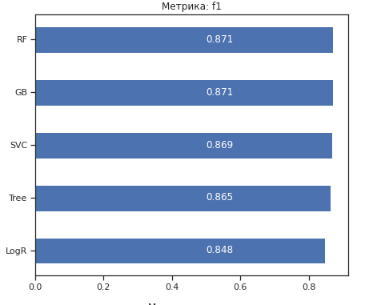
Лучшая модель по метрике precision: Логическая регрессия



Лучшая модель по метрике recall: Случайный лес



Лучшая модель по метрике f1: Метод опорных векторов



Лучшая модель по метрике ROC AUC: Случайный лес



Вывод: на основании четырех метрик, лучшей оказались построенная на методе опорных векторов модель и модель случайного леса.

# **Заключение**

Из всех рассмотренных алгоритмов: "Logistic Regression", "Support vector machine", "Decision tree", "Gradient boosting", "Random forest" для модели определения победителя матча лучших игроков по дисциплине League of Legends наиболее эффективными оказались алгоритм случайного леса и метод опорных векторов. Как известно Random forest борется с переобучением модели, следовательно можно сделать вывод о том, что датасет является довольно разрозненным, поэтому другие методы могли привести к возникновению проблемы переобучения, а "Random forest" успешно обошёл эту проблему.

# **Список литературы**

1. Репозиторий курса "Технологии машинного обучения", бакалавриат, 6 семестр. Лекции по теории машинного обучения. Ю.Е. Гапанюк [Электронный ресурс]. – Электрон. дан. - URL: <https://github.com/ugapanyuk/ml_course_2020/wiki/COURSE_TMO> (дата обращения: 31.05.2020)
2. League Of Legends High elo Ranked Games (2020) [Электронный ресурс]. – Электрон. дан. - URL: <https://www.kaggle.com/gyejr95/league-of-legends-challenger-ranked-games2020?select=Challenger_Ranked_Games.csv> (дата обращения: 04.06.2020)
3. Машинное обучение (часть 1). А.М.Миронов [Электронный ресурс]. – Электрон. дан. - URL:

<http://www.intsys.msu.ru/staff/mironov/machine_learning_vol1.pdf> (дата обращения: 31.05.2020)

1. Scikit learn[Электронный ресурс]. – Электрон. дан. - URL: <https://scikit-learn.org/stable/index.html> (дата обращения: 31.05.2020)