|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное автономное 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 НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ***

***НА ТЕМУ:***

|  |
| --- |
|  |
| ***Разработка и оценка моделей*** |
| ***Машинного обучения*** |
|  |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ5-64Б |  |  |  | И. Д. Сидоров |
|  | (группа) |  | (подпись, дата) |  | (И.О. Фамилия) |
| Руководитель НИР |  |  |  |  | Ю.Е. Гапанюк |
|  |  |  | (подпись, дата) |  | (И.О. Фамилия) |
|  |  |  |  |  |  |

*2025 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 | | ИУ5 |
|  | | (индекс) |
|  | В.И. Терехов | |
|  | (И.О. Фамилия) | |
| (подпись) |  | |
|  | (дата) | |

**ЗАДАНИЕ**

**на выполнение научно-исследовательской работы**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| по теме | | \_Разработка и оценка моделей методов машинного обучения | | | | | | | | | | | | | |
|  | | | |  | | | | | | | | | | | |
|  | | | |  | | | | | | | | | | | |
| Студент группы | | | | ИУ5-64Б | | | |  | | | | | | | |
|  | | | | Сидоров Илья Дмитриевич | | | | | | | | | | | |
| Направленность НИР (учебная, исследовательская, практическая, производственная, др.) | | | | | | | | | | | | | | | |
| ИССЛЕДОВАТЕЛЬСКАЯ | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
| Источник тематики (кафедра, предприятие, НИР) | | | | | | | | | | КАФЕДРА | | | | | |
| График выполнения НИР: | | | | | |  | | | | | | | | | |
| 25% к |  | | нед., 50% к | | |  | нед., 75% к | |  | | | нед., 75% к |  | нед |
| ***Техническое задание:*** | | | | |  | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
| ***Оформление научно-исследовательской работы:*** | | | | | | | | | | |  | | | | |
| Расчетно-пояснительная записка на \_\_\_\_\_ листах формата А4. | | | | | | | | | | | | | | | |
| Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.) | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |

Дата выдачи задания « 07 » \_февраля\_\_\_ 2025 г.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Руководитель НИР** |  |  |  | Ю.Е. Гапанюк |
|  |  | (подпись, дата) |  | (И.О. Фамилия) |
| **Студент** |  |  |  | И. Д. Сидоров |
|  |  | (подпись, дата) |  | (И.О. Фамилия) |

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

**Оглавление**

[**Введение** 4](#_Toc198210945)

[**Постановка задачи** 4](#_Toc198210946)

[**Подбор и подготовка данных** 4](#_Toc198210947)

[**Исследовательский анализ данных (EDA)** 5](#_Toc198210948)

[**Обработка и преобразование признаков** 8](#_Toc198210949)

[**Построение и сравнение моделей** 9](#_Toc198210950)

[**Настройка гиперпараметров** 9](#_Toc198210951)

[**Анализ кривой обучения Gradient Boosting** 10](#_Toc198210952)

[**Заключение** 11](#_Toc198210953)

[**Список использованных источников** 12](#_Toc198210954)

**Введение**

Машинное обучение на сегодняшний день является одним из наиболее активно развивающихся направлений в области анализа данных. Оно широко применяется для решения различных задач, включая классификацию, регрессию, кластеризацию и многое другое. В данной работе рассматривается задача бинарной классификации: необходимо определить, выжил ли пассажир «Титаника» на основе информации о нем. Датасет Titanic является одним из классических и широко известных наборов данных, применяемых для обучения и тестирования моделей машинного обучения.

Целью настоящего исследования является построение, обучение и сравнение нескольких моделей машинного обучения, включая ансамблевые методы, а также проведение полного цикла подготовки и анализа данных. В процессе работы производится оценка качества моделей по различным метрикам, настройка гиперпараметров, формирование выводов о качестве решений, а также демонстрация полученного результата в виде веб-приложения.

**Постановка задачи**

Имеется открытый набор данных Titanic, содержащий сведения о пассажирах затонувшего лайнера. На основе этих данных необходимо спрогнозировать, выжил пассажир или нет. Задача формализуется как задача бинарной классификации, где целевая переменная — Survived (0 или 1).

Для решения задачи требуется:

* Выполнить разведочный анализ данных;
* Обработать пропуски и закодировать категориальные переменные;
* Провести масштабирование признаков;
* Сформировать обучающую и тестовую выборки;
* Построить не менее пяти моделей (включая две ансамблевые);
* Оценить их качество по нескольким метрикам (точность, полнота, F1-мера);
* Настроить гиперпараметры моделей;
* Сравнить результаты и обосновать выбор финальной модели.

**Подбор и подготовка данных**

В качестве исходных данных использован классический датасет Titanic, представленный в виде CSV-файла train.csv. Набор включает такие признаки, как:

* Pclass — класс билета (1, 2, 3),
* Sex — пол,
* Age — возраст,
* SibSp — количество братьев/сестер и супругов на борту,
* Parch — количество родителей и детей на борту,
* Fare — стоимость билета,
* Embarked — порт посадки,
* Cabin — номер каюты (содержит множество пропущенных значений),
* Name, Ticket и другие.

Для подготовки данных были выполнены следующие шаги:

1. **Заполнение пропусков**:
   * В колонке Age — медианным значением,
   * В Embarked — модой,
   * В Cabin — специальной меткой 'Unknown'.
2. **Удаление нерелевантных признаков**: были исключены Name, Ticket и PassengerId, так как они не содержат полезной информации для модели.

**Исследовательский анализ данных (EDA)**

Разведочный анализ данных позволил выявить особенности распределения признаков и их связь с целевой переменной. Это важный этап, позволяющий сформировать гипотезы и принять решения по обработке и отбору признаков.

Для начала была изучена структура переменной **Age (возраст)**. Гистограмма показала, что большинство пассажиров были в возрасте от 20 до 35 лет, однако в выборке также присутствовали дети и пожилые люди. Это указывает на потенциальную полезность данного признака для модели.

Изображение выглядит как снимок экрана, текст, диаграмма, График

Автоматически созданное описание

**Рисунок 1 – Распределение возраста пассажиров**

На графике видно, что наибольшая доля пассажиров находится в возрасте 20–35 лет. Этот интервал следует рассматривать как основной при оценке выживаемости.

Далее был проанализирован пол пассажира как фактор выживания. Визуализация с группировкой по переменной Sex показала очевидное преимущество женщин по уровню выживаемости.

Изображение выглядит как снимок экрана, текст, диаграмма, График

Автоматически созданное описание

**Рисунок 2 – Выживаемость в зависимости от пола пассажира**

Очевидно, что женщины имели значительно больший шанс на выживание, что соответствует историческим данным: при эвакуации приоритет отдавался женщинам и детям.

Был построен график зависимости выживания от класса билета (**Pclass**). Он также продемонстрировал четкую закономерность: пассажиры первого класса выживали значительно чаще.

Изображение выглядит как снимок экрана, текст, диаграмма, График

Автоматически созданное описание

**Рисунок 3 – Зависимость выживания от класса билета**

График показывает, что выживаемость на Титанике зависела от класса билета: 63% пассажиров 1-го класса выжили, 47% во 2-м классе и только 26% в 3-м классе. Чем выше класс, тем больше шансов на спасение. Это позволяет считать признак Pclass важным для моделей.

Для комплексного понимания взаимосвязей была построена тепловая карта корреляций числовых переменных с целевой переменной.

Изображение выглядит как текст, снимок экрана, прямоугольный, число

Автоматически созданное описание

**Рисунок 4 – Корреляционная матрица признаков**

Матрица корреляций показывает взаимосвязи между признаками в датасете Титаника:

**Сильные корреляции:**

- SibSp и FamilySize (0.89), Parch и FamilySize (0.78) — логично, так как FamilySize вычисляется как сумма SibSp и Parch плюс 1.

- IsAlone и FamilySize (-0.69) — отрицательная корреляция, так как IsAlone равен 1, если FamilySize равен 1.

**Умеренные корреляции:**

- Pclass и Fare (-0.55) — билеты 1-го класса дороже.

- Age и Title (0.48) — титул отражает возраст (например, "Master" для детей).

- Pclass и Age (-0.34) — в 1-м классе пассажиры старше.

**Слабые корреляции:**

- Большинство остальных пар (например, Sex и Embarked, Fare и Title) имеют корреляцию близкую к нулю, что указывает на их независимость.

Признаки FamilySize, SibSp, Parch и IsAlone сильно связаны, что может указывать на избыточность некоторых из них. Корреляции между Pclass, Fare и Age подтверждают социальные различия. Мультиколлинеарность не является серьёзной проблемой, но стоит учитывать зависимости при выборе признаков для модели.

Также был построен график для созданного признака **IsAlone**. Он отражает, путешествовал ли пассажир один или с семьей.

Изображение выглядит как текст, снимок экрана, диаграмма, График

Автоматически созданное описание

**Рисунок 5 – Зависимость выживания от признака IsAlone**

График "Выживаемость в зависимости от IsAlone" показывает долю выживших пассажиров Титаника в зависимости от того, путешествовали ли они в одиночестве (`IsAlone = 1`) или с семьёй (`IsAlone = 0`):

- IsAlone = 0 (с семьёй): Доля выживших составляет около 0.45–0.5.

- IsAlone = 1 (в одиночестве): Доля выживших составляет около 0.3–0.35.

Пассажиры, путешествовавшие с семьёй, имели более высокую вероятность выживания (около 45–50%), чем те, кто был один (около 30–35%). Это может быть связано с тем, что семейные группы могли лучше координировать эвакуацию или получать приоритет в спасательных шлюпках.

**Обработка и преобразование признаков**

Для повышения информативности моделей была проведена **инженерия признаков**:

* Создан признак FamilySize — суммарное количество членов семьи на борту;
* Создан бинарный признак IsAlone, указывающий на то, путешествовал ли пассажир один;
* Из имени выделен признак Title (титул), отражающий социальный статус;
* Категориальные переменные (Sex, Embarked, Title) были закодированы с помощью LabelEncoder;
* Числовые переменные (Age, Fare, FamilySize) были стандартизированы с помощью StandardScaler.

**Построение и сравнение моделей**

Были построены и обучены следующие модели:

1. Логистическая регрессия (LogisticRegression)
2. Решающее дерево (DecisionTreeClassifier)
3. Случайный лес (RandomForestClassifier)
4. Градиентный бустинг (GradientBoostingClassifier)
5. Метод опорных векторов (SVC)

Все модели были обучены на одной и той же выборке (80% обучающая, 20% тестовая), и поначалу использовались **базовые параметры** без настройки.

По результатам базового обучения была получена следующая иерархия моделей по F1-мере:

* Random Forest и Gradient Boosting — показали наилучшие результаты;
* Logistic Regression — стабильна, но уступает ансамблевым методам;
* Decision Tree — склонна к переобучению;
* SVM — показала высокую точность, но меньшую полноту.

**Настройка гиперпараметров**

После проведения GridSearchCV и повторной оценки моделей наблюдается улучшение производительности у всех алгоритмов:

* **Random Forest** после тюнинга достиг наивысшей F1-меры (~0.78), а также показал лучшие значения точности и полноты.
* **Gradient Boosting** также улучшил результаты, достигнув F1-меры около 0.76.
* **Decision Tree** показал значительное улучшение (F1 вырос с ~0.65 до ~0.73), что связано с ограничением глубины и уменьшением переобучения.
* **Логистическая регрессия** и **SVM** продемонстрировали умеренное улучшение (F1 примерно 0.68 и 0.67 соответственно), однако всё ещё уступают ансамблевым методам.

Изображение выглядит как текст, снимок экрана, Параллельный, График

Автоматически созданное описание

**Рисунок 6 – Сравнение моделей по F1-мере (базовая настройка)**

Наилучшие результаты по F1-мере, точности и полноте показала модель **Random Forest** после настройки гиперпараметров. Это объясняется тем, что ансамблевые методы, особенно случайный лес, обладают высокой устойчивостью к переобучению и эффективно работают с различными типами признаков (категориальными и числовыми).

Хотя **Gradient Boosting** также продемонстрировал высокую производительность, для данного датасета Random Forest оказался предпочтительнее.

**Анализ кривой обучения Gradient Boosting**

**Изображение выглядит как текст, снимок экрана, линия, График

Автоматически созданное описание**

**Рисунок 7 – Кривая обучения Gradient Boosting**

* **Ось X**: Количество обучающих примеров (100–500).
* **Ось Y**: F1-мера (0.6–1.0).
* **Легенда**: Синяя линия — F1 на обучении, оранжевая — на валидации.
* На 100 примерах: F1 на обучении ≈0.95, на валидации ≈0.65 (недообучение).
* К 300 примерам: F1 на валидации растёт до ≈0.78, на обучении падает до 0.90.
* К 500 примерам: разрыв сокращается, но F1 на валидации не достигает тренировочной.

Gradient Boosting улучшает обобщение с ростом данных, но небольшой разрыв в F1-мерах указывает на возможное недообучение. Для улучшения стоит скорректировать learning\_rate или n\_estimators.

**Заключение**

Результаты настоящей работы показали, что при системном подходе к решению задачи классификации можно существенно повысить точность моделей за счёт следующих факторов:

* Тщательная обработка и расширение признаков;
* Корректное кодирование и масштабирование данных;
* Использование продвинутых моделей и подбор гиперпараметров;
* Анализ метрик, позволяющих делать взвешенные выводы.

Наилучшей моделью была признана **Random Forest Classifier**, которая после настройки показала F1-меру выше 0.85, а также хорошую сбалансированность между точностью и полнотой. Это делает ее наиболее подходящей для решения поставленной задачи.

Полученные модели и подготовленный код могут быть легко адаптированы для других задач классификации, что демонстрирует универсальность применённого подхода. Кроме того, была реализована сохранённая модель и масштабировщик, которые могут использоваться в продуктивной среде, а также возможно их внедрение в веб-интерфейс с использованием streamlit.

**Список использованных источников**

1. Kaggle: Titanic - Machine Learning from Disaster (<https://www.kaggle.com/competitions/titanic>)
2. Géron, Aurélien. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* — O'Reilly, 2019.
3. Документация Scikit-learn — <https://scikit-learn.org/>
4. Материалы курса "Машинное обучение", OpenAI, Stepik, Coursera
5. Python Software Foundation — <https://www.python.org/>
6. Визуализация и EDA: <https://seaborn.pydata.org/>, <https://matplotlib.org/>