МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное образовательное
учреждение высшего образования
ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Институт математики, механики и компьютерных наук
имени И. И. Воровича

Направление подготовки 02.03.02 — «Фундаментальная информатика и информационные технологии»

Кулишов Е. С., 3 курс, 4 группа

Курсовая работа

РЕШЕНИЕ ЗАДАЧИ БИНАРНОЙ КЛАССИФИКАЦИИ МЕТОДАМИ МАШИННОГО ОБУЧЕНИЯ

Научный руководитель:
А. В. Абрамян

_____ _____
оценка (рейтинг) подпись руководителя

Ростов-на-Дону
2025

Оглавление

Введение

На сегодняшний день невозможно представить общество без широкого использования технологий, которые стали неотъемлемой частью повседневной жизни. Одним из таких достижений является SMS-сообщение, которое значительно упростило коммуникацию и общение между людьми. Мы используем SMS не только для общения с близкими и коллегами, но и для получения различных уведомлений, например, для подтверждения банковских операций или получения информации от государственных огранизаций. На личном опыте могу подтвердить важность SMS сообщений, ведь без них я бы не смог оперативно узнавать важную информацию (например, перенос даты экзамена или напоминание о мероприятии от старосты группы). Отмечу, что SMS-сообщения активно применяются и в сфере безопасности: уведомления о плохих погодных условиях (что свойственно для области, в которой я живу), предупреждения о чрезвычайных ситуациях (такие как высокая пожароопасность, что тоже характерно для моего региона) и другие важные сообщения приходят именно через этот канал.

Однако, несмотря на очевидную полезность данного средства связи, оно не лишено проблем. Одной из самых острых является проблема спама, когда пользователи сталкиваются с нежелательными сообщениями, часто содержащими мошеннические схемы, фишинг-атаки или вредоносные ссылки [7]. На собственном опыте скажу, что такие сообщения не только раздражают, но и представляют серьезную угрозу безопасности, особенно когда речь идет о личных данных, например, данных банковских карт. В особенной зоне риска находятся пожилые люди, которые (к сожалению) не владеют современными технологиями на базовом уровне, что часто приводит к потере их средств или даже имущества из-за мошенников.

Эффективность машинного обучения в решении задачи классификации SMS-сообщений лежит в том, что оно учитывает особенности коротких текстов, таких как использование сокращений, орфографических ошибок, эмодзи и смешанных языков, что требует внимательной предварительной

обработки данных и выбора правильных методов векторизации текста. Именно поэтому в данной работе рассматриваются методы, такие как Naive Bayes, Logistic Regression и Random Forest, которые зарекомендовали себя как эффективные для подобного рода задач.

Практическая значимость данной работы обусловлена важностью борьбы с SMS-спамом (который процветает не только у нас в регионе, но и по всей стране), который представляет собой реальную угрозу для пользователей, особенно в моей области (Ростовская область), где мошенничество в сфере мобильной связи продолжает набирать масштабы. Работа направлена на создание системы, которая бы эффективно фильтровала спам и помогала пользователям защищаться от угроз в мобильной среде.

Предобработка данных

В моём исследовании предварительная обработка SMS реализована с использованием специального реализованного мной класса TextPreprocessor. Он наследуется от базовых классов scikit-learn BaseEstimator и TransformerMixin: это позволяет легко интегрировать его в конвейеры обработки данных [2]. Основная функциональность класса сосредоточена вокруг нескольких ключевых операций по очистке и нормализации текста.

```
class TextPreprocessor(BaseEstimator, TransformerMixin):

    def __init__(self):

        self.stop_words = set(stopwords.words('english'))

        self.lemmatizer = WordNetLemmatizer()
```

Первым шагом в процессе предварительной обработки является удаление всех неалфавитных символов с помощью регулярного выражения re.sub(r'[^a-zA-Z]', ' ', text). Эта операция удаляет из текста цифры, знаки препинания, специальные символы и другие несущественные элементы, которые могут помешать анализу содержимого сообщения. Важно отметить, что этот подход намеренно сохраняет пробелы между словами, заменяя удаленные символы пробелами, а не пустой строкой, чтобы предотвратить слипание слов при последующей обработке (что было замечено мной на практике). После этого текст приводится к нижнему регистру с помощью text.lower(). Это в свою очередь позволяет избежать дублирования одних и тех же слов в разных регистрах.

Следующим шагом является разбиение текста на отдельные слова с помощью метода split(), после чего применяются сразу две важные операции нормализации. Во-первых, используя предварительно загруженный список стоп-слов из NLTK (stopwords.words («английский»)), я удаляю из текста определённого SMS распространенные слова, которые не несут

существенного смыслового значения (артикли, предлоги, местоимения и т. д.). Во-вторых, остальные слова лемматизируются с помощью NLTK WordNetLemmatizer, который возвращает словам их базовую форму (например, «running» → «run», «better» → «good»). Лемматизация, в отличие от более простого стемминга, учитывает морфологию слов и возвращает их нормальную (словарную) форму, что повышает точность дальнейшего анализа.

```
    def clean_text(self, text):
        text = re.sub(r'[^a-zA-Z]', ' ', text)
        text = text.lower()
        words = text.split()
        words = [self.lemmatizer.lemmatize(word)
                 for word in words if word not in
self.stop_words]
        return ' '.join(words)
```

Реализация предварительной обработки в виде класса с использованием методов fit и transform обеспечивает совместимость с scikit-learn API и упрощает включение этого этапа в конвейеры обработки данных [2]. В этом случае метод fit не выполняет никаких обучающих операций, поскольку для предварительной обработки не требуются параметры, извлеченные из данных, а просто возвращает self в соответствии с интерфейсом (см. семантику ЯП Python). Основная работа выполняется в методе transform, который применяет функцию clean_text к каждому элементу входных данных.

```
    def fit(self, X, y=None):
        return self
```

```
    def transform(self, X, y=None):
        return [self.clean_text(text) for text in X]
```

Особенностью этого подхода является его ориентация на английский язык, что проявляется в использовании английского списка стоп-слов и лемматизатора WordNet. Для обработки сообщений на других языках потребуется соответствующая адаптация — замена списка стоп-слов и использование других языковых моделей для лемматизации. Отмечу, что текущая реализация не включает в себя ещё некоторые возможные этапы предварительной обработки, такие как исправление опечаток, обработка сленга и сокращений или выделение именованных сущностей, которые при необходимости могут быть добавлены в существующую структуру классов.

Используемые методы машинного обучения

Для решения задачи бинарной классификации SMS-сообщений на спам и не-спам были выбраны пять различных алгоритмов машинного обучения, каждый из которых обладает особыми характеристиками, которые важны для работы с текстовыми данными.

Логистическая регрессия — это один из самых простых методов классификации, который тем не менее часто даёт отличные результаты, даже когда дело касается сложных наборов данных. В нашем случае она показала свою эффективность, несмотря на довольно высокую размерность данных. Этот метод основывается на вычислении вероятности принадлежности объекта к определённому классу с помощью логистической функции. Важно отметить, что логистическая регрессия остаётся довольно интерпретируемой, что позволяет легче понимать, какие признаки (например, частота использования определённых слов) влияют на решение модели. Для задачи классификации SMS-сообщений этот метод хорош тем, что он легко справляется с линейными зависимостями между признаками и метками классов, например, когда определённые слова или фразы чаще встречаются в спам-сообщениях.

Случайный лес — это уже более сложный метод, который использует сразу несколько деревьев решений для повышения точности классификации. В отличие от простых деревьев решений, случайный лес строит множество деревьев на случайных подмножествах данных и признаков, что значительно уменьшает риск переобучения. Этот метод является достаточно мощным, поскольку автоматически определяет важность признаков, не требуя предварительного отбора, что особенно важно для работы с текстовыми данными. В рамках данной работы случайный лес показал хорошие результаты, сочетающие высокую точность и устойчивость к шуму в данных.

XGBoost представляет собой современную реализацию градиентного бустинга, который в последние годы стал одним из самых популярных методов для решения задач машинного обучения. В отличие от других методов, XGBoost строит деревья решений последовательно, при этом каждое новое дерево исправляет ошибки предыдущих. Такой подход позволяет достичь высокой точности, особенно в случае сложных зависимостей между признаками и метками классов. XGBoost оказался полезным инструментом для классификации SMS-сообщений, поскольку он способен учесть нелинейные связи между словами и их контекстом.

Метод опорных векторов (SVM) с линейным ядром был выбран для задачи классификации по причине его способности работать с высокоразмерными данными, что является характерной особенностью текстовых наборов данных после применения таких методов, как TF-IDF. SVM находит оптимальную гиперплоскость, которая разделяет классы с максимальной точностью. Этот метод, несмотря на свою сложность, обладает высокой производительностью, особенно когда размерность признаков значительно превышает количество примеров в обучающей выборке. Также, благодаря параметру probability=True, SVM может вычислять вероятности для каждого класса, что полезно для анализа уверенности модели в своих предсказаниях.

Наивный байесовский классификатор использует теорему Байеса для оценки вероятности принадлежности сообщения к одному из классов. Он основывается на предположении, что признаки (слова в сообщении) независимы, что в реальности редко бывает верно, но, несмотря на это, метод часто показывает хорошие результаты. Мультиномиальный вариант наивного Байеса особенно хорошо работает с текстовыми данными, поскольку использует частотные характеристики слов. Этот метод отличается быстрой обработкой больших объемов данных и хорошей масштабируемостью, что также было важным фактором при его применении в данной задаче.

Для оценки эффективности моделей использовалась метрика F1-score, которая представляет собой гармоническое среднее точности и полноты и особенно полезна для задач с несбалансированными классами, как в нашем случае. Мы разделили данные на обучающую и тестовую выборки с фиксированным random_state, чтобы обеспечить воспроизводимость результатов.

Далее применялась векторизация текста с помощью метода TF-IDF с ограничением на 5000 наиболее часто встречающихся слов. Все модели обучались и оценивались на основе этой векторизации, после чего была выбрана лучшая модель, показавшая наилучшие результаты по метрике F1-score.

Результаты и их анализ

Проведённое мной исследование по классификации SMS-сообщений на спам и не-спам на открытом датасете с платформы Kaggle позволило мне получить ряд важных и полезных результатов, анализ которых представлен ниже. Основные выводы сделаны на основе трех ключевых визуализаций результатов исследования: распределения классов в используемом датасете (см. рисунок 1 в Приложении), матрицы ошибок (см. Рисунки 3–7 в Приложении) и распределения длины сообщений (см. рисунок 2 в Приложении).

Анализ распределения классов

График распределения классов (см. Рисунок 1 в Приложении) демонстрирует значительный дисбаланс в исходных данных: количество легитимных сообщений (ham) существенно превышает количество спама (spam). Этот дисбаланс видов сообщений в датасете создает определенные сложности для обучения выбранных мной моделей, так как алгоритмы могут проявлять склонность к предсказанию класса большинства. Для достижения объективных результатов при оценке качества моделей в моём проекте особое внимание уделялось метрикам, устойчивым к дисбалансу классов, таким как F1-score и полнота (recall) для класса спама. В Приложении на Рисунке 1 приведена визуализация в виде гистограммы распределения сообщений по классам.

Анализ матриц ошибок каждой модели

Изучив матрицу ошибок каждой использованной модели, я пришёл к выводу, что лучшая модель (по балансу точности и полноты) является SVM (или метод опорных векторов). При 100% точности классификации модель сохраняет максимальный recall (53.5%) и минимальное количество ложных срабатываний (FP = 0). Визуализация матрицы ошибок данной модели приведена в Приложении как Рисунок 5.

Логистическая регрессия же допускает высокую специфичность – способность идентифицировать обычные сообщения (в моём случае) –

(99.8%), почти не показывая ложных срабатываний. Также она имеет относительно низкий recall (42.2%) для спама (эта модель пропускает значительную часть нежелательных сообщений). Визуализация матрицы ошибок данной модели приведена в Приложении как Рисунок 3.

Случайный лес показал себя как более точная версия логистической регрессии (100% точность и улучшенный 47.6% recall), но при этом всё ещё не идеальная модель в плане классификации SMS сообщений. Визуализация матрицы ошибок данной модели приведена в Приложении как Рисунок 4.

XGBoost можно охарактеризовать как что-то среднее между SVM и логистической регрессией, ведь у неё точность составляет 99.3% и средний recall, составляющий 45.1%. Модель немного хуже случайного леса, но сохраняет баланс между точностью и полнотой. Визуализация матрицы ошибок данной модели приведена в Приложении как Рисунок 7.

Наивный Байес же показал себя как менее всего подходящую модель для решения задачи бинарной классификации SMS, ведь несмотря на 100% точность он имеет 37.7% recall, что является худшим показателем среди тестируемых моделей. Визуализация матрицы ошибок данной модели приведена в Приложении как Рисунок 6.

Анализ распределения длины сообщений

График распределения длины сообщений (см. Рисунок 2 в Приложении) показывает заметные различия между двумя классами:

- Ham-сообщения в среднем короче и имеют более компактное распределение по длине.
- Spam-сообщения демонстрируют более широкий разброс длин с выраженным «хвостом» в область длинных сообщений.

Это наблюдение согласуется с практикой спамеров, которые часто используют большие тексты для рекламы или мошеннических предложений, в то время как обычные сообщения (например, уведомления или личная переписка) обычно более короткие. В Приложении на Рисунке 2 приведена подробная диаграмма распределения длины сообщений.

Общие выводы

1. Качество классификации показывает, что почти все модели эффективно фильтруют легитимные сообщения, но некоторые из них нуждаются в доработке для лучшего выявления спама, особенно его новых и сложных форм.

2. Длина сообщения является полезным признаком, который может улучшить качество классификации при правильном использовании.

3. Ложные срабатывания минимальны (всего 2 модели показали ложное срабатывание, которое при этом осталось минимальным у них обеих), что очень важно для практического применения системы — пользователи почти не столкнутся с ситуацией, когда важное сообщение ошибочно попадет в спам.

Заключение

В ходе выполнения этой курсовой работы я решил актуальную задачу бинарной классификации SMS-сообщений на спам и не-спам с использованием методов машинного обучения. Процесс проделанной работы включал: подготовку данных, очистку и векторизацию до выбора и обучения моделей. Таким образом я смог выполнить поставленную перед собой задачу, используя готовые библиотеки и модели.

Из всех моделей, которые я тестировал, наилучшие результаты показал метод опорных векторов (SVM). Эта модель продемонстрировала высокую точность и почти идеальную работу по выявлению легитимных сообщений, при этом минимизируя количество ложных срабатываний (вплоть до 0 ложных срабатываний). Это особенно важно, потому что пользователи не будут сталкиваться с ситуациями, когда важные сообщения окажутся в спаме, что может быть критически важным для многих.

Процесс анализа длины сообщений также оказался полезным. Я заметил, что спам-сообщения обычно длиннее и более разнообразны по длине, чем обычные сообщения, что подтверждает общую закономерность: спам часто содержит больше текста, так как это связано с рекламными предложениями и мошенническими схемами. Этот признак может быть использован для улучшения классификации, если добавить его в качестве дополнительного параметра.

Что касается практической ценности проведённой работы, то она очевидна: внедрение предложенной системы фильтрации SMS-сообщений может значительно повысить защиту пользователей от мошенничества, которое распространяется через мобильные устройства.

Мой проект может служить как учебный пример применения машинного обучения для решения задач бинарной классификации. Код моего проекта выложен на GitHub и находится в открытом доступе [6].

Литература

1. Используемый набор данных с его описанием [Электронный ресурс] – URL: https://www.kaggle.com/datasets/mariumfaheem666/spam-sms-classification-using-nlp/data
2. Scikit-learn официальная документация [Электронный ресурс] – URL: https://scikit-learn.org/stable/
3. XGBoost официальная документация [Электронный ресурс] – URL: https://xgboost.readthedocs.io/
4. Jurafsky D., Martin J.H. Speech and Language Processing (3rd Edition). – 2020. [Электронный ресурс] – URL: https://web.stanford.edu/~jurafsky/slp3/
5. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. [Электронный ресурс] — URL: https://www.nltk.org/book/
6. Ссылка на мой проект в GitHub. [Электронный ресурс] – URL: https://github.com/Dilray/Course-paper
7. Самая распространённая проблема спама – СМС спам. [Электронный ресурс] – URL: https://tenchat.ru/media/2919598-sms-spam-opasnosti-i-sposoby-zaschity
8. Финансовые потери от спам-смс в 2023–2024 году. [Электронный ресурс] – URL: https://ptsecurity.com/ru-ru/research/analytics/phishing-attacks-on-organizations-in-2022-2023/

Приложения

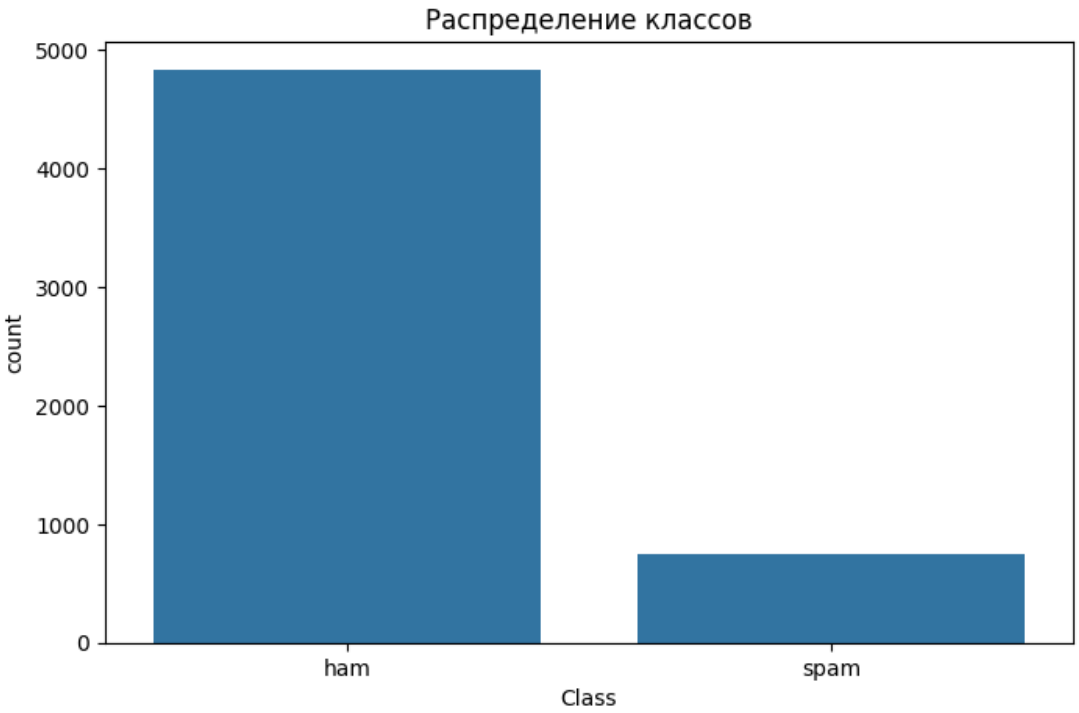

Рисунок 1. Распределение сообщений по классам.

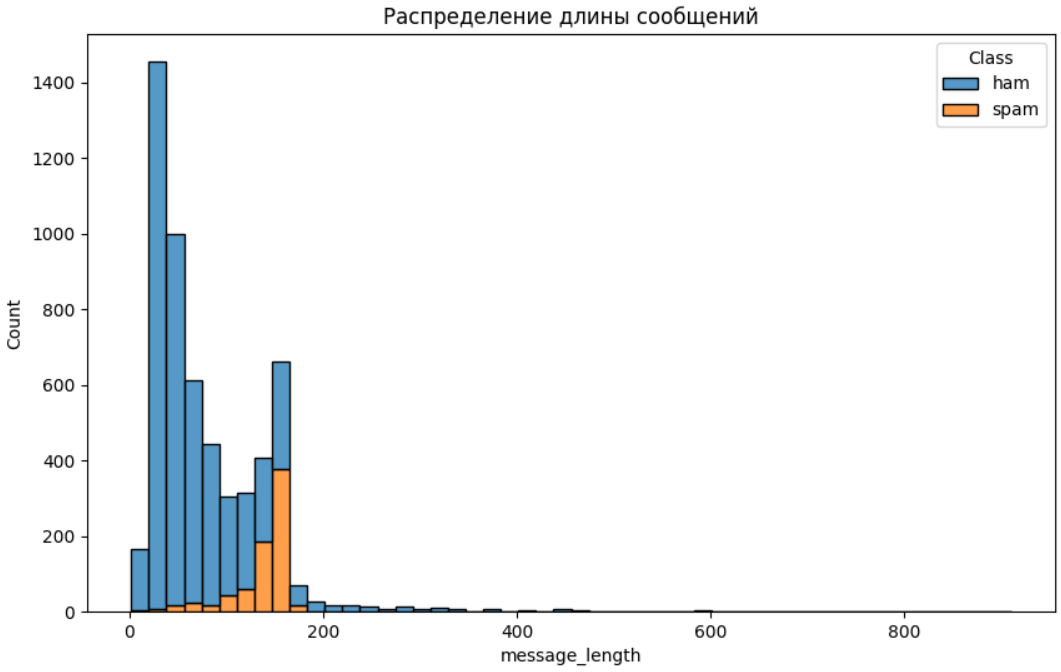

Рисунок 2. Диаграмма распределения длины сообщений.

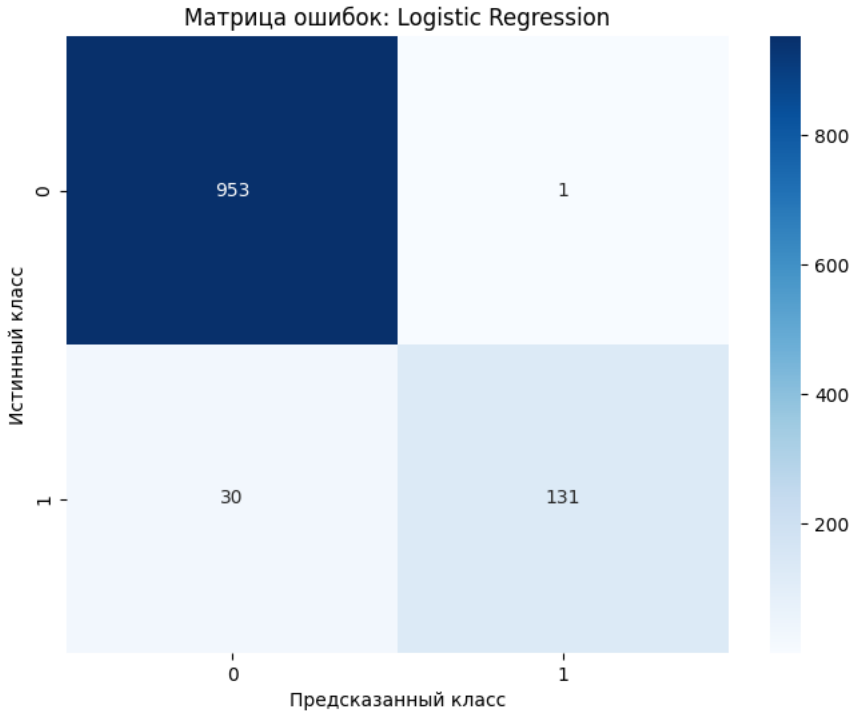

Рисунок 3. Матрица ошибок Логической регрессии.

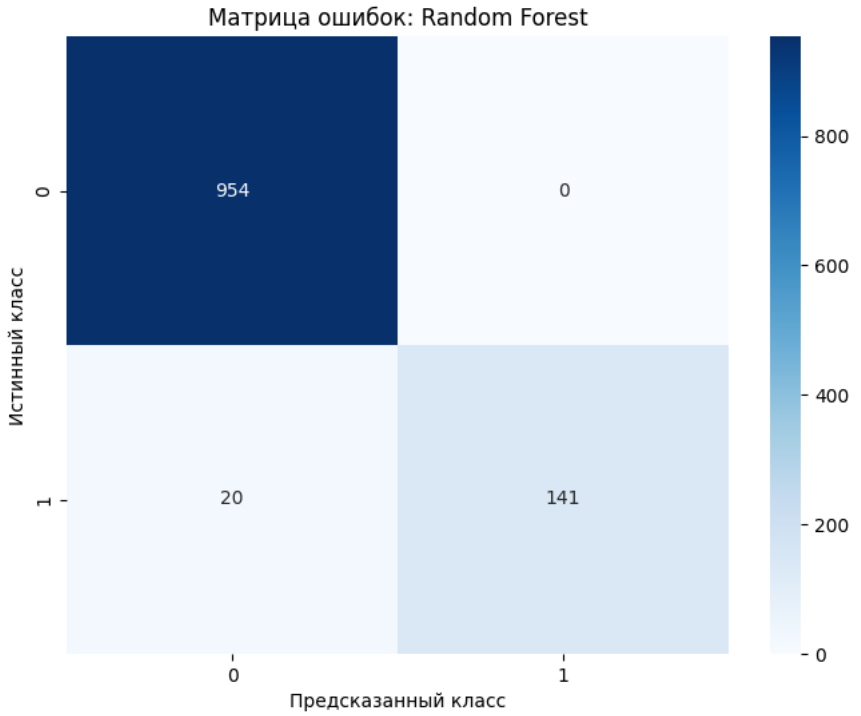

Рисунок 4. Матрица ошибок Случайного леса.

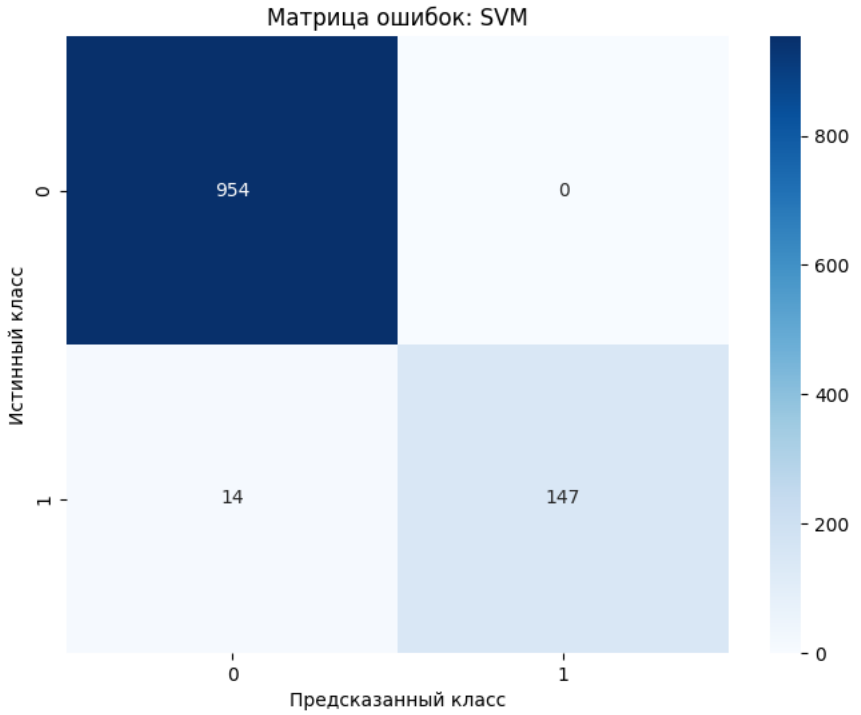

Рисунок 5. Матрица ошибок Метода опорных векторов.

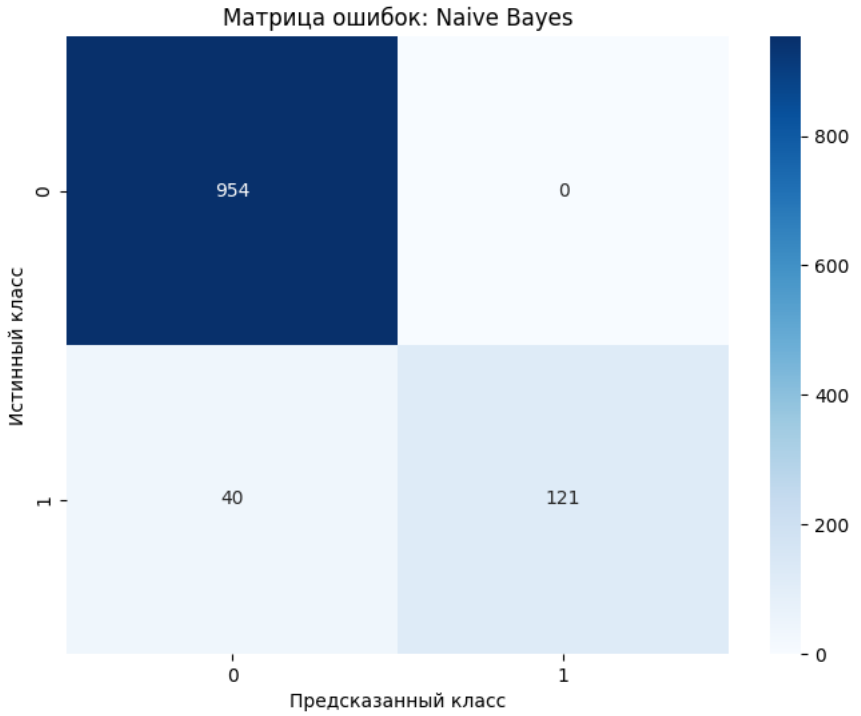

Рисунок 6. Матрица ошибок Наивного Байеса.

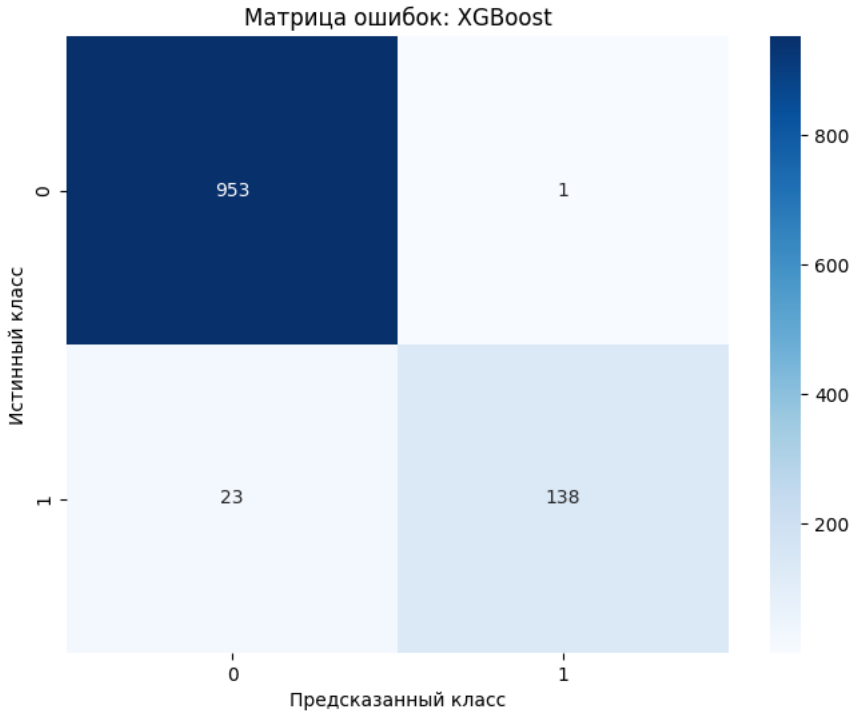

Рисунок 7. Матрица ошибок XGBoost.

Модель	Точность (Precision)	Полнота (Recall)	F1-score	Accuracy
Logistic regression	0.992424	0.813665	0.894198	0.972197
Random forest	1	0.875776	0.933775	0.982063
XGBoost	0.992806	0.857143	0.92	0.978475
SVM	1	0.913043	0.954545	0.987444
Naïve Bayes	1	0.751553	0.858156	0.964126

Таблица 1. Сравнение метрик используемых моделей.