

Пояснительная записка.

Промежуточная аттестация.

Тема: «Предсказание возможности поломки фрезерного станка»

Цель работы: Разработка и тестирование приложения для оценки рисков нарушения работы фрезерного станка и выявления возможности его поломки.

Автор: Билалов Радмир Рамисович

Датасет:

«AI4I 2020 Predictive Maintenance Dataset - UCI Machine Learning Repository»
(DOI: 10.24432/C5HS5C)

- <https://archive.ics.uci.edu/dataset/601/ai4i+2020+predictive+maintenance+dataset>

GitHub:

- https://github.com/Rad2white/innopolis-predicting_machine_failure

Содержание

| | |
|---|----|
| 1. Анализ предметной области и обзор существующих решений. | 3 |
| 1.1. Исследование предметной области..... | 5 |
| 1.2. Анализ результатов измерений | 7 |
| 2. Проектирование системы..... | 12 |
| 2.1. Анализ требований | 12 |
| 2.2. Выбор инструментов для разработки | 13 |
| 2.3. Технология основного приложения | 14 |
| 2.4. Интерфейс основного приложения | 15 |
| 2.5. Набор классических алгоритмов для обработки данных..... | 18 |
| 2.6. Инструмент для построения отчетности | 21 |
| 3. Заключение..... | 22 |

1. Анализ предметной области и обзор существующих решений.

Оборудование является ключевым элементом в любом производственном или бизнес-процессе, и его неполадки могут иметь серьезные последствия, такие как финансовые убытки, задержки в производстве и ухудшение качества продукции или услуг. Прогнозирование отказов оборудования становится критически важным, поскольку это позволяет заранее выявлять возможные проблемы и принимать меры для их предотвращения или минимизации негативных воздействий на бизнес.

Существует несколько подходов к решению этой задачи:

- Сбор данных в реальном времени: Использование IoT-сенсоров и других устройств для непрерывного мониторинга работы оборудования. Это помогает выявить предвестия проблем и принять меры еще до того, как они достигнут критической стадии.
- Применение машинного обучения: Использование алгоритмов и моделей для анализа данных о работе оборудования и предсказания вероятности отказа. Это позволяет более точно оценивать риски и предпринимать меры для их снижения.
- Регулярное техническое обслуживание: Предотвращение отказов, связанных с неправильным обслуживанием и износом, через систематическое техническое обслуживание.
- Автоматизированный мониторинг состояния оборудования: Использование систем мониторинга для обнаружения отклонений в работе оборудования и предупреждения операторов о возможных проблемах.
- Облачные сервисы для хранения и анализа данных: Использование облачных решений для централизованного хранения и анализа данных, что обеспечивает доступность данных из различных источников и применение сложных алгоритмов анализа.
- Виртуализация: Эмуляция работы оборудования и симуляция сценариев отказа для тестирования и оптимизации систем до их внедрения в производственной среде.
- Разработка протоколов восстановления: Создание процедур для быстрого восстановления работы после отказа с целью минимизации времени простоя и уменьшения убытков.
- Обучение персонала: Подготовка персонала по эксплуатации и обслуживанию оборудования с целью снижения вероятности человеческих ошибок.
- Программные решения для управления активами: Использование специализированных программных продуктов для автоматизации управления активами и пассивными компонентами производственного процесса.

Интеграция существующих систем:

Интеграция систем мониторинга и прогнозирования с уже существующими ERP (Enterprise Resource Planning) и CRM (Customer Relationship Management) системами представляет собой стратегически важный шаг для обеспечения комплексного обзора бизнес-процессов и операционной деятельности предприятия.

Этот процесс предоставляет следующие преимущества:

- Целостный обзор бизнес-процессов: Интеграция данных о состоянии оборудования с системами ERP позволяет создать единое информационное пространство, в котором бизнес-лидеры могут анализировать и оценивать не только состояние технических средств, но и влияние их работы на ключевые бизнес-процессы.

- Оптимизация управленческих решений: Объединение данных о прогнозах отказов с информацией из CRM системы позволяет принимать управленческие решения на основе всестороннего анализа. Например, можно эффективнее планировать ресурсы, управлять запасами и оптимизировать производственные процессы.

- Улучшение клиентского опыта: Интеграция с CRM системой дает возможность отслеживать влияние состояния оборудования на обслуживание клиентов. Это позволяет предсказывать и предотвращать возможные сбои в обслуживании, что, в свою очередь, улучшает общий клиентский опыт.

- Автоматизация отчетности: Интеграция данных мониторинга и прогнозирования с ERP системой обеспечивает автоматическую генерацию отчетов о состоянии оборудования и его влиянии на производственные показатели. Это сокращает ручной труд по формированию отчетов и повышает точность предоставляемой информации.

- Повышение эффективности и экономии: Совместное использование данных из различных систем позволяет выявлять скрытые корреляции и оптимизировать процессы более эффективным образом. Это приводит к сокращению времени простоя оборудования, увеличению общей производительности и снижению затрат на техническое обслуживание.

В общем, эта интеграция обеспечивает не только лучший контроль над состоянием оборудования, но и позволяет предприятию более адаптивно реагировать на изменения внутри и вне компании, создавая более устойчивое и эффективное производственное окружение.

1.1. Исследование предметной области

Поскольку реальные наборы данных по прогнозированию технического обслуживания обычно трудно получить и особенно трудно опубликовать, нам был предоставлен синтетический набор данных, который отражает реальное прогнозирование технического обслуживания, встречающееся в промышленности.

Данный датасет содержит данные о состоянии оборудования и его техническом обслуживании, собранные в течение нескольких месяцев. Он предназначен для задачи классификации на основе машинного обучения. В нем содержатся 10000 записей, хранящихся в виде строк с 14 признаками в столбцах.

Каждая точка данных содержит следующие параметры:

- UID (уникальный идентификатор): Нумерация от 1 до 10000.
 - ID продукта: Составлен из букв L, M или H, представляя низкое (50%), среднее (30%) и высокое (20%) качество продукта соответственно, а также серийный номер.
 - Тип продукта: L, M или H.
 - Температура воздуха [K]: Генерируется с использованием стохастических алгоритмов и нормируется на стандартное отклонение 2 K в районе 300 K.
 - Температура процесса [K]: Генерируется с использованием стохастических алгоритмов, нормированными на стандартное отклонение 1 K, и добавляется к температуре воздуха, увеличенной на 10 K.
 - Скорость вращения [об/мин]: Рассчитывается на основе мощности 2860 Вт с добавлением нормально распределенного шума.
 - Крутящий момент [Нм]: Значения нормально распределены вокруг 40 Нм с стандартным отклонением 10 Нм и не имеют отрицательных значений.
 - Износ инструмента [мин]: Значения износа для разных вариантов качества продукта (H/M/L) добавляются к времени использования инструмента в процессе работы.
 - Метка "Отказ станка": Указывает, произошел ли отказ станка в данной точке данных и какой режим отказа активирован.
- Отказ станка включает следующие режимы:
- Отказ по износу инструмента (TWF): Замена инструмента в случайный момент времени между 200 и 240 минутами (120 раз в наборе данных).
 - Отказ теплоотвода (HDF): Сбой процесса при разнице температур ниже 8,6 K и скорости вращения ниже 1380 об/мин (115 случаев).
 - Отказ по мощности (PWF): Сбой процесса, если мощность ниже 3500 Вт или выше 9000 Вт (95 случаев).

- Отказ из-за перегрузки (OSF): Сбой процесса при превышении производства износа инструмента и крутящего момента порогового значения (98 случаев).
- Случайные отказы (RNF): Вероятность отказа 0,1% для каждого процесса (5 случаев).

Если хотя бы один из режимов отказа активен, процесс считается неудачным, и устанавливается метка "Машинный отказ" в 1. Этот синтетический набор данных моделируется на основе существующего фрезерного станка и предоставляет информацию для прогнозирования технического обслуживания в промышленных процессах.

В нашем датасете для прогнозирования мы использовали данные без учета режимов отказа станков (9 признаков), принимая во внимание исключительно метку «Отказ станка». Однако, в исходном датасете представлена колонка «*Failure Type*», в которой указана категория отказа станка.

Исходный датасет представлен на рисунке 1.

| | UDI | Product ID | Type | Air temperature [K] | Process temperature [K] | Rotational speed [rpm] | Torque [Nm] | Tool wear [min] | Target | Failure Type |
|---|-----|------------|------|---------------------|-------------------------|------------------------|-------------|-----------------|--------|--------------|
| 0 | 1 | M14860 | M | 298.1 | 308.6 | 1551 | 42.8 | 0 | 0 | No Failure |
| 1 | 2 | L47181 | L | 298.2 | 308.7 | 1408 | 46.3 | 3 | 0 | No Failure |
| 2 | 3 | L47182 | L | 298.1 | 308.5 | 1498 | 49.4 | 5 | 0 | No Failure |
| 3 | 4 | L47183 | L | 298.2 | 308.6 | 1433 | 39.5 | 7 | 0 | No Failure |
| 4 | 5 | L47184 | L | 298.2 | 308.7 | 1408 | 40.0 | 9 | 0 | No Failure |

Рис.1 – Исходный датасет

1.2. Анализ результатов измерений

В процессе проведения исследования был осуществлен анализ данных с использованием различных визуализаций и статистических методов. Результаты EDA представлены в следующих таблицах и графиках:

- Описательная статистика:

Таблица, созданная с использованием функции `df.describe()`, предоставляет обзор основных статистических характеристик каждого признака в наборе данных. Это включает в себя среднее значение, стандартное отклонение, минимальное и максимальное значения, медиану и квантили.

| | udi | type | airtemperature | processtemperature | rotationalspeedrpm | torquenm | toolwearmin | target | failuretype |
|-------|--------------|--------------|----------------|--------------------|--------------------|--------------|--------------|--------------|--------------|
| count | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 |
| mean | 5000.500000 | 1.800600 | 27.854930 | 37.855560 | 1538.776100 | 39.986910 | 107.951000 | 0.033900 | 1.105100 |
| std | 2886.895680 | 0.600230 | 2.000259 | 1.483734 | 179.284096 | 9.968934 | 63.654147 | 0.180981 | 0.628883 |
| min | 1.000000 | 1.000000 | 23.150000 | 33.550000 | 1168.000000 | 3.800000 | 0.000000 | 0.000000 | 1.000000 |
| 25% | 2500.750000 | 1.000000 | 26.150000 | 36.650000 | 1423.000000 | 33.200000 | 53.000000 | 0.000000 | 1.000000 |
| 50% | 5000.500000 | 2.000000 | 27.950000 | 37.950000 | 1503.000000 | 40.100000 | 108.000000 | 0.000000 | 1.000000 |
| 75% | 7500.250000 | 2.000000 | 29.350000 | 38.950000 | 1612.000000 | 46.800000 | 162.000000 | 0.000000 | 1.000000 |
| max | 10000.000000 | 3.000000 | 32.350000 | 41.650000 | 2886.000000 | 76.600000 | 253.000000 | 1.000000 | 6.000000 |

Рис. 2 – Описание числовых параметров

- Диаграммы рассеяния:

Функция `sns.pairplot(df)` создает набор диаграмм рассеяния для всех пар признаков. Это позволяет визуально оценить зависимости между переменными и выявить возможные паттерны в данных.

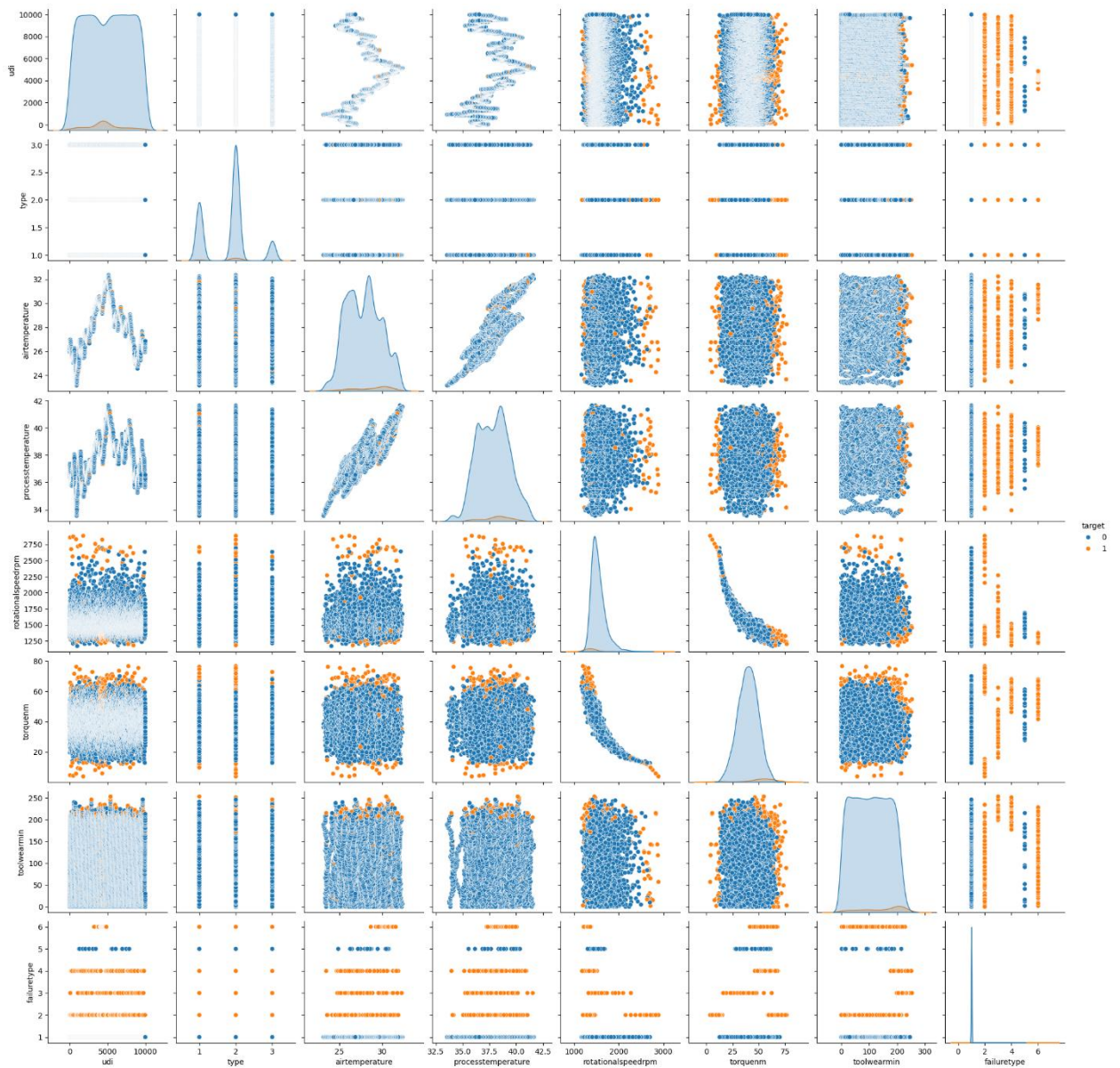


Рис. 3 – Диаграмма рассеяния (парная диаграмма)

- Тепловая карта корреляции:

Тепловая карта, построенная с использованием `sns.heatmap(df)`, визуализирует корреляцию между признаками. Это позволяет выявить потенциальные взаимосвязи между переменными и оценить степень их влияния друг на друга.

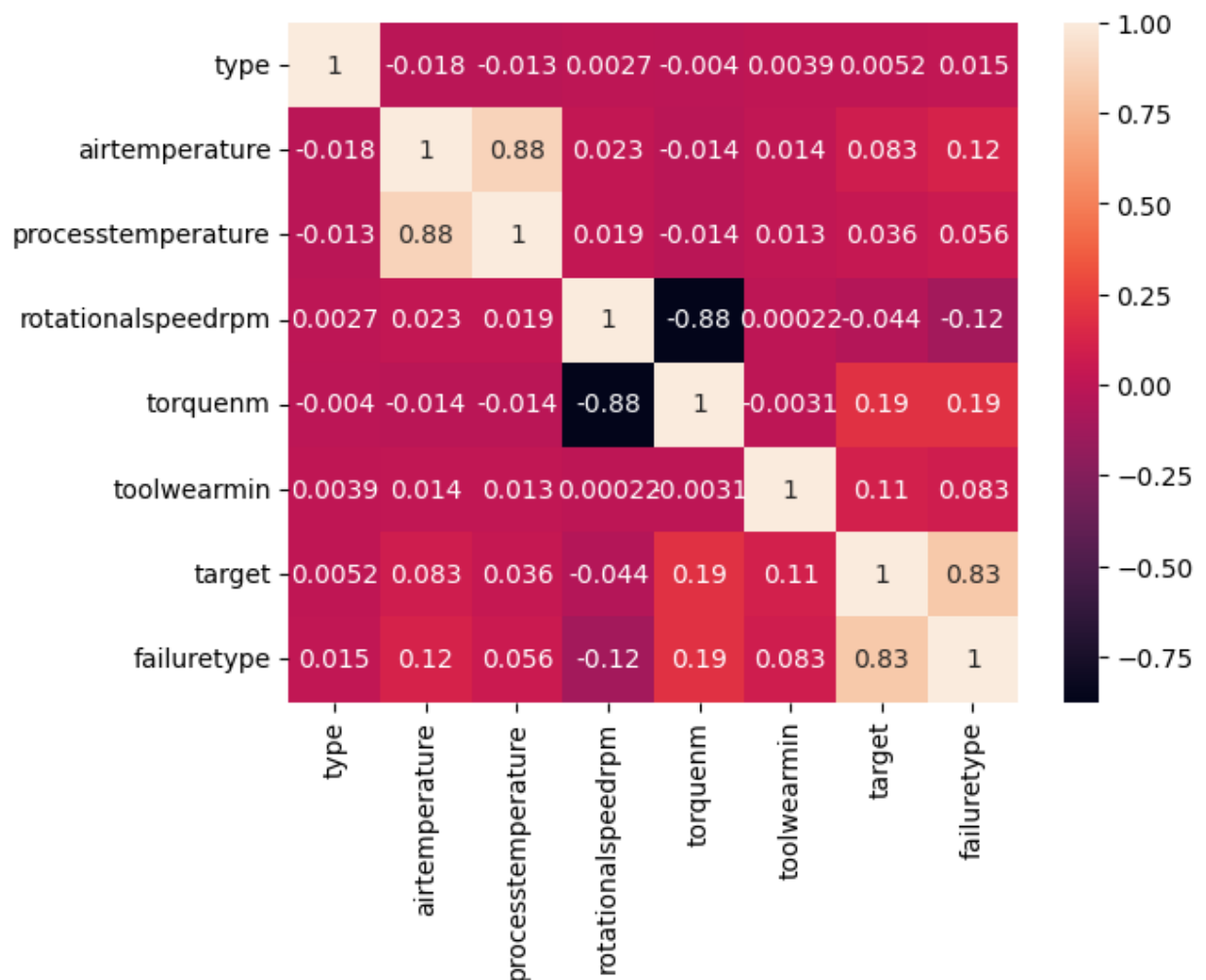


Рис. 4 – Матрица корреляции (тепловая карта)

- Матрица пропущенных значений:

Результаты функции `msno.matrix(df)` представлены в матрице, отображающей распределение пропущенных значений в данных. Это важный инструмент для выявления пропусков и понимания их распределения в различных признаках.

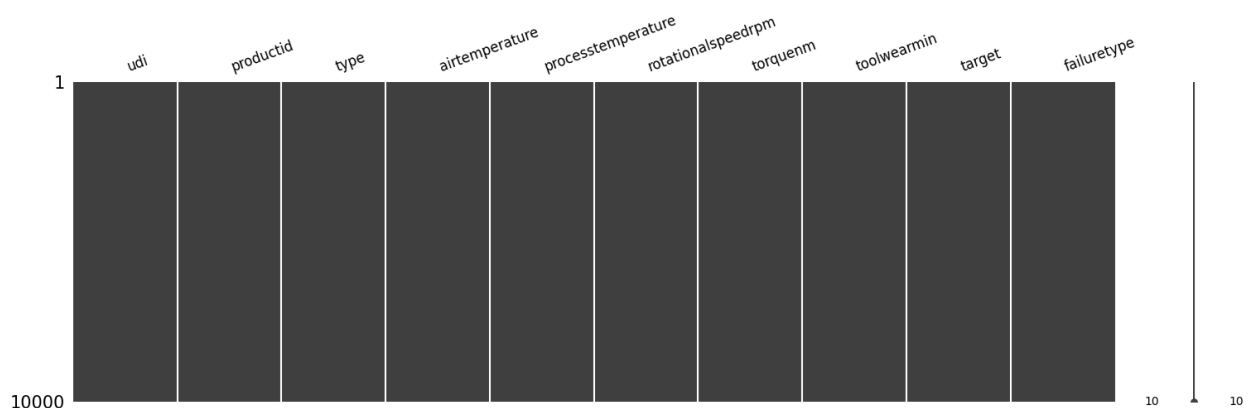


Рис. 5 – • Матрица пропущенных значений

- Распределение типов станков:

Представлены результаты анализа распределения типов станков в наборе данных. Это включает в себя график распределения и круговую диаграмму, которые позволяют легко воспринимать соотношение между различными типами станков.

Таблица 1 – Мапинг типа станка.

| Тип станка | Числовой код |
|------------|--------------|
| М | 1 |
| L | 2 |
| H | 3 |

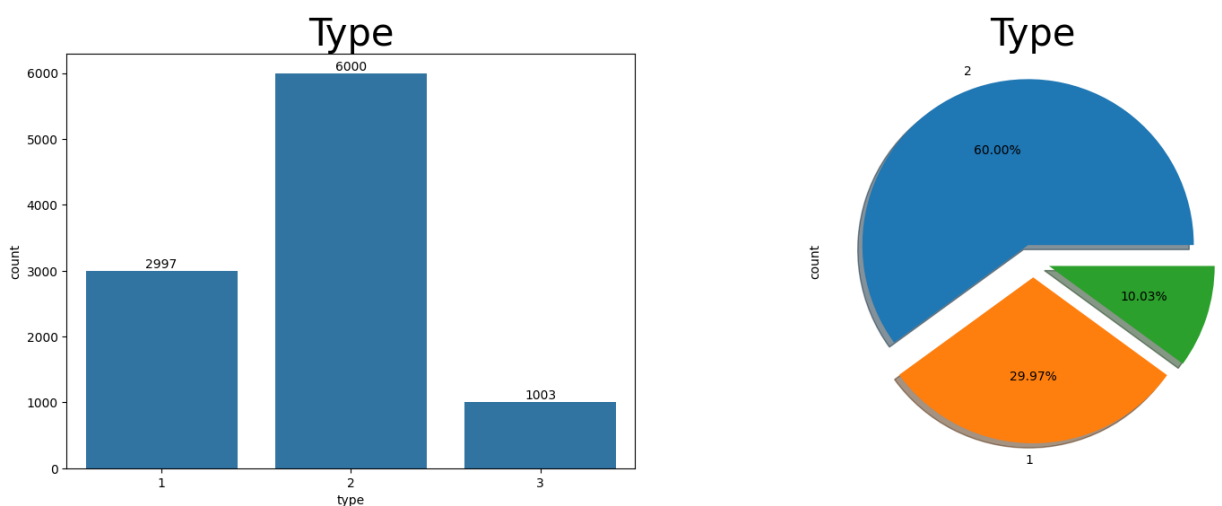


Рис. 6 – Распределение типов станков

- Распределение целевого параметра:

Представлены результаты анализа распределения целевого параметра, связанного с вероятностью поломки станка. Визуализации включают график распределения и круговую диаграмму, что облегчает понимание структуры целевой переменной в контексте всего датасета.

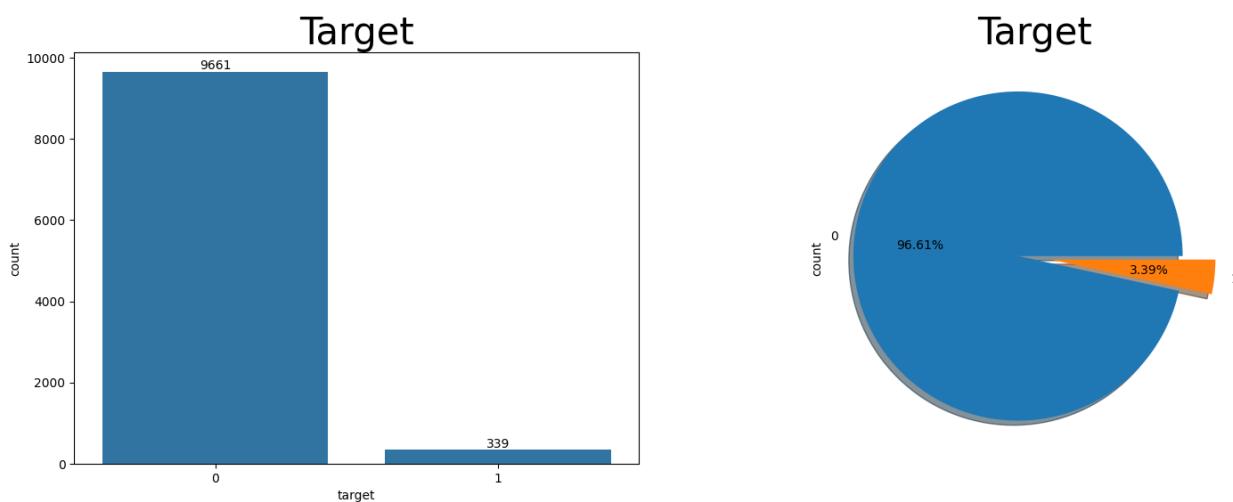


Рис. 7 – Распределение целевого параметра

Комплексный анализ результатов измерений (EDA) на основе этих инструментов обеспечивает более глубокое понимание структуры данных, идентификацию потенциальных проблемных областей и формулирование дальнейших шагов для улучшения качества и использования данных в дальнейших этапах проекта.

Визуализации распределений добавляют контекст к основному анализу и позволяют быстро выявить особенности в данных, связанные с типами станков и целевым параметром. Полученные результаты помогут принять во внимание дополнительные аспекты при разработке модели прогнозирования поломок станков. К примеру, так как целевой параметр – это выявление возможности поломки, то из распределения целевого параметра четко видно, что количество вышедших из строя станков низкое. Ввиду этого факта, для оценки модели была выбрана метрика *recall* (полнота):

$$\text{Recall} = \frac{TP}{TP + FN}$$

Интуитивно метрика показывает долю найденных нарушений работы станка из всех релевантных. Чем меньше ложно отрицательных срабатываний, тем выше *recall* модели.

2. Проектирование системы

2.1. Анализ требований

1. Выбор библиотеки для веб-приложения:

Проект будет реализован с использованием Flask, легковесного и гибкого фреймворка для создания веб-приложений на языке программирования Python.

В качестве альтернативного варианта были рассмотрены библиотеки: Django, известный своей полнотой и комплексностью, и Streamlit, специализированный для создания интерактивных веб-приложений для анализа данных. Однако, данные библиотеки неактуальны для текущей задачи и датасета.

2. Стабильная работа приложения:

Важное требование - приложение должно обеспечивать стабильную работу без вызова исключений и ошибок в процессе использования.

3. Основные страницы приложения:

Необходимо реализовать, как минимум, «Главную страницу» приложения, которая предоставит пользователю доступ к основным функциям и прогнозу модели.

4. Выбор базы данных:

Использование базы данных SQLite обусловлено легкостью интеграции и минимальной настройкой.

В качестве альтернативного варианта было рассмотрено использование PostgreSQL для проектов с более сложными потребностями в хранении данных. Однако, данный вариант стал неактуален исходя из задачи и датасета.

5. Управление версиями и репозиторий:

Исходный код проекта будет размещен в репозитории на github.com, обеспечивая централизованное управление версиями и совместную работу.

6. requirements.txt и ReadMe.md:

Файл requirements.txt будет содержать перечень всех библиотек проекта и их версий, упрощая процесс установки необходимых зависимостей.

Файл ReadMe.md в корневой папке проекта предоставит пользователям подробное описание приложения, его возможностей и инструкции по использованию.

2.2. Выбор инструментов для разработки

- Язык программирования:

Для разработки веб-интерфейса и обработки данных будет использоваться язык программирования Python, который отлично подходит для работы с фреймворком Flask.

- Система контроля версий:

Git будет использоваться для отслеживания изменений в коде и обеспечения коллективной работы. Репозиторий будет размещен на платформе github.com для удобства совместной разработки.

- Среда разработки:

Visual Studio Code (VSCode) выбран в качестве среды разработки для комфортной и эффективной работы разработчиков, которая обеспечивает хороший опыт разработки и отладки.

- Текстовый редактор для файлов разметки:

Для создания и редактирования файлов разметки, таких как ReadMe.md, будет использоваться удобный текстовый редактор, например, VSCode.

2.3. Технология основного приложения

Основное приложение разрабатывается с использованием фреймворка Flask и языка программирования Python 3.11. Принято решение использовать следующие технологии и компоненты для построения и функционирования веб-интерфейса:

- Фреймворк для веб-приложения:

Flask выбран в качестве основного веб-фреймворка из-за своей простоты, гибкости и легковесности. Он предоставляет минимальный набор инструментов, позволяя разработчикам эффективно создавать веб-приложения.

- Библиотеки для машинного обучения:

XGBoost будет использован в качестве модели для прогнозирования вероятности поломки станка. Эта библиотека хорошо зарекомендовала себя в задачах классификации. Однако, в процессе разработки были рассмотрены и другие модели в качестве альтернативы. Данные модели реализованы в Jupyter Notebook.

- СУБД:

База данных SQLite выбрана для хранения данных, так как она легко интегрируется с Flask и не требует сложной конфигурации. Она подходит для небольших проектов и обеспечивает простоту использования.

Технологии были выбраны с учетом требований проекта, обеспечивая надежность, производительность и удобство разработки и использования.

Библиотеки и версии для веб-приложения:

- flask==3.0.0
- pandas==2.1.2
- joblib==1.3.2
- datetime==5.2

Используемый процессор: Intel® Core™ i5-8250U @ 3.40GHz.

- Дополнительные библиотеки:

Дополнительные библиотеки будут использоваться для обработки данных, анализа и предобработки данных перед обучением модели. Они используются в Jupyter Notebook.

Библиотеки и версии для обучения модели:

- matplotlib==3.8.0
- seaborn==0.13.0
- category_encoders==2.6.3
- missingno==0.5.2
- sklearn==1.3.0

2.4. Интерфейс основного приложения

Веб-интерфейс реализован с помощью Flask

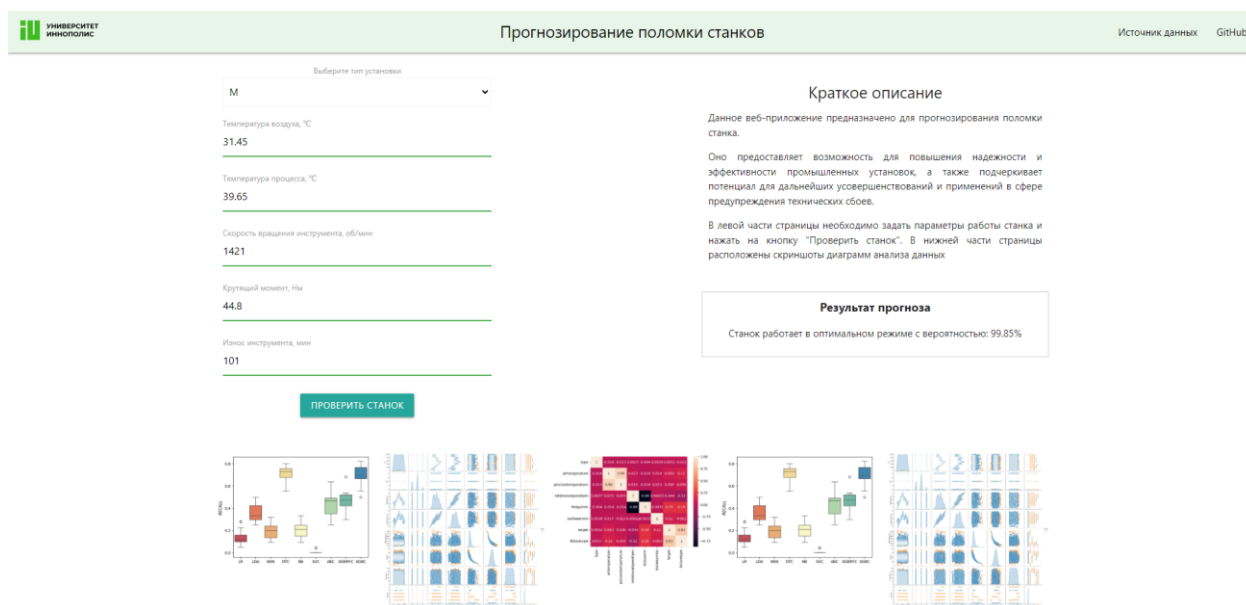


Рис.8 – Веб-интерфейс приложения

Интерфейс основного приложения разрабатывается с упором на интуитивную навигацию, понятность для конечного пользователя и эффективную визуализацию результатов работы модели прогнозирования поломки станка. Веб-интерфейс основного приложения включает следующие элементы:

- Главная страница:

На главной странице представлена краткая информация о приложении и его функционале. Здесь пользователь может ознакомиться с общим описанием приложения и получить краткую инструкцию по использованию.

- Поля ввода данных:

Интерфейс предоставляет функционал для ввода данных, необходимых для предсказания возможности поломки станка.

Выберите тип установки

M

Температура воздуха, °C

31.45

Температура процесса, °C

39.65

Скорость вращения инструмента, об/мин

1421

Крутящий момент, Нм

44.8

Износ инструмента, мин

101

ПРОВЕРИТЬ СТАНОК

Рис.9 – Поля ввода данных

- Предсказание вероятности поломки:

Разработана страница для выполнения предсказаний модели XGBoost. После ввода данных пользователь может запросить предсказание вероятности поломки станка (нажатием на кнопку).

- Визуализация EDA:

Для наглядной демонстрации проведенного анализа данных используются графики и диаграммы. В нижней части страницы расположена визуализация ключевых признаков, а также представление общей статистики.

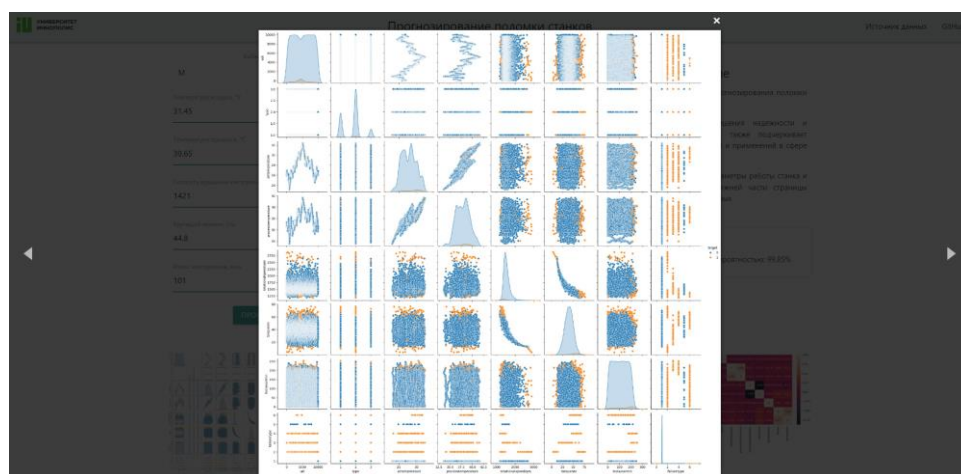


Рис.10 – Визуализация EDA

- Страница для просмотра результатов:

Результаты выводятся на главной странице в читаемой форме.

Краткое описание

Данное веб-приложение предназначено для прогнозирования поломки станка.

Оно предоставляет возможность для повышения надежности и эффективности промышленных установок, а также подчеркивает потенциал для дальнейших усовершенствований и применений в сфере предупреждения технических сбоев.

В левой части страницы необходимо задать параметры работы станка и нажать на кнопку "Проверить станок". В нижней части страницы расположены скриншоты диаграмм анализа данных

Результат прогноза

Станок работает в оптимальном режиме с вероятностью: 99.85%

Рис.11 – Просмотр результатов

- Простота использования:

Интерфейс приложения разрабатывается с акцентом на простоту использования, что позволяет пользователям без особых навыков в области анализа данных и машинного обучения эффективно взаимодействовать с приложением.

Интерфейс основного приложения создается с учетом принципов удобства использования, чтобы предоставить пользователям интуитивно понятный и эффективный инструмент для работы с моделью прогнозирования поломки станка.

2.5. Набор классических алгоритмов для обработки данных

В процессе анализа и выбора наилучшей модели для задачи прогнозирования поломки станка был проведен обширный анализ различных классических алгоритмов машинного обучения. Были использованы следующие модели:

- Logistic Regression:

Логистическая регрессия была использована для линейного моделирования зависимости между признаками и вероятностью поломки станка.

- Linear Discriminant Analysis (LDA):

LDA применялся для моделирования различий между классами данных и выделения наиболее значимых признаков.

- K-Nearest Neighbors (KNN):

KNN использовался для классификации данных на основе их схожести с ближайшими соседями в пространстве признаков.

- Decision Tree Classifier:

Решающее дерево было применено для построения структуры решений на основе иерархии признаков.

- Gaussian Naive Bayes (GNB):

Модель GNB использовалась для оценки вероятности поломки станка, основываясь на предположении о нормальном распределении признаков.

- Support Vector Classifier (SVC):

SVC был использован для поиска гиперплоскости, которая наилучшим образом разделяет классы данных в пространстве.

- AdaBoost Classifier:

AdaBoost применялся для адаптивного комбинирования результатов отдельных моделей с целью улучшения общей производительности.

- XGBRFClassifier и XGBClassifier:

Исследование также включало в себя две модели градиентного бустинга: XGBRFClassifier (экстремальные случайные леса) и XGBClassifier (градиентный бустинг).

Анализ проводился с использованием метода кросс-валидации (k-fold) с целью оценки производительности моделей на различных подмножествах данных.

Таблица 2. Показатели метрики *recall* для кросс-валидации (k-fold).

| | <i>recall</i> (mean) | <i>recall</i> (std) |
|---------------|----------------------|---------------------|
| LR | 0.13958 | -0.068483 |
| LDA | 0.358548 | -0.07898 |
| KNN | 0.193091 | -0.068317 |
| DTC | 0.688141 | -0.061996 |
| NB | 0.204267 | -0.067244 |
| SVC | 0.005929 | -0.01512 |
| ABC | 0.428198 | -0.097765 |
| XGBRFC | 0.473423 | -0.084323 |
| XGBC | 0.693972 | -0.087969 |

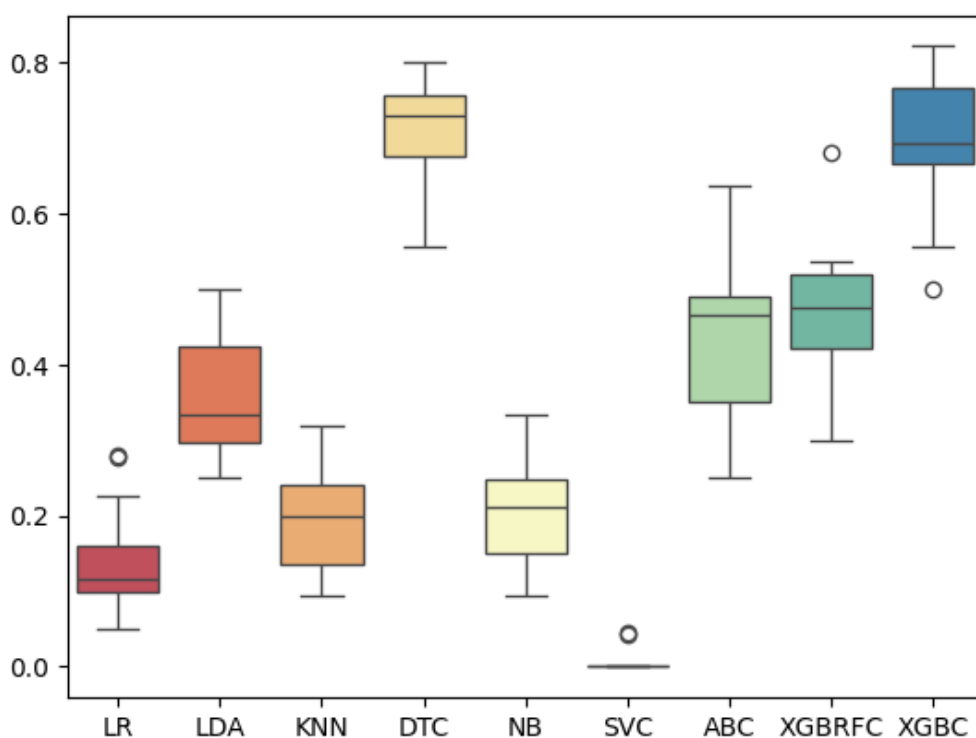


Рис. 12 – Сравнение алгоритмов по метрике *recall*

В результате данного исследования наилучшей моделью оказался XGBClassifier, демонстрирующий наилучшую метрику $recall = 0.69$, что подтверждает его эффективность в прогнозировании поломок станка в данном контексте. Эта модель была выбрана для интеграции в веб-интерфейс приложения для прогнозирования вероятности поломки станка. Оценка итоговой модели XGBClassifier представлена в таблице 3.

Таблица 3. Метрики итоговой модели XGBClassifier.

| <i>Accuracy</i> | <i>Precision</i> | <i>Recall</i> | <i>F1-Score</i> | <i>ROC AUC Score</i> |
|-----------------|------------------|---------------|-----------------|----------------------|
| 0.985 | 0.788 | 0.672 | 0.726 | 0.833 |

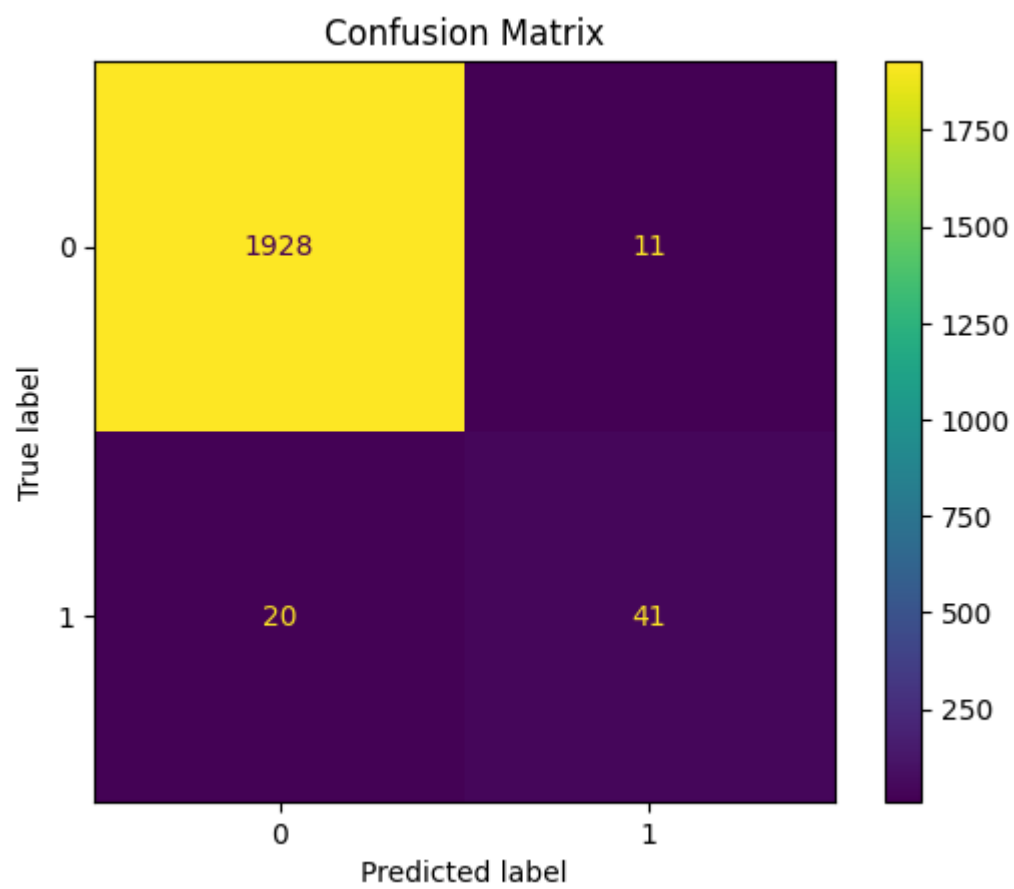


Рис. 14 – Confusion-матрица для выбранной модели (XGBClassifier)

2.6. Инструмент для построения отчетности

Для информативной отчетности о состоянии станка после ввода значений признаков и нажатия кнопки "Проверить станок" был разработан эффективный механизм, направленный представление результата.

- Страница результатов:

После выполнения проверки, пользователь получает сообщение в контейнере, где представлена информация о возможности поломки станка в зависимости от введенных признаков.

- Уведомление о состоянии:

На странице результатов выводится текст, сообщающий о текущем состоянии станка. Например, "Станок подвержен поломке с вероятностью...", или "Станок работает в оптимальном режиме с вероятностью...".

- Вероятностная оценка:

Важным элементом отчетности является вероятностная оценка состояния станка. Эта информация помогает пользователям лучше понимать степень риска или безопасности текущей ситуации.

3. Заключение

Разработанный проект по прогнозированию поломки станка и созданию веб-интерфейса на его основе представляет собой важный шаг в области предупреждения технических сбоев в промышленных процессах.

- Достижения:

Успешная реализация веб-интерфейса на основе модели XGBoost позволяет оперативно оценивать вероятность поломки станка, что имеет важное значение для поддержания бесперебойной работы оборудования.

- Технологические решения:

Выбор библиотеки Flask для создания веб-интерфейса обеспечивает легкость в разработке и интеграции с моделью прогнозирования.

- Надежность и безопасность:

Приложение разработано с учетом стабильности и безопасности. Оно не вызывает исключений и ошибок при работе, что обеспечивает надежную функциональность.

- Дальнейшие перспективы:

Данный проект может служить основой для дальнейших исследований и улучшений. Возможные направления включают в себя расширение функционала, оптимизацию работы с базой данных, а также улучшение пользовательского интерфейса.

- Вклад в индустрию:

Проект вносит свой вклад в область промышленной безопасности и эффективности производственных процессов, обеспечивая оперативное предсказание возможных сбоев и снижая риски простоев оборудования.

Заключение подчеркивает важность разработанного приложения для повышения надежности и эффективности промышленных установок, а также подчеркивает потенциал для дальнейших усовершенствований и применений в сфере предупреждения технических сбоев.