

Научная статья

Original article

УДК 004.891.2

doi: [https://doi.org/10.55186/2413046X\\_2026\\_11\\_6\\_82](https://doi.org/10.55186/2413046X_2026_11_6_82)

edn: QUMIQF

**ВЛИЯНИЕ ХАРАКТЕРИСТИК РАЗРАБОТКИ НА КАЧЕСТВО  
ПРОГНОЗИРОВАНИЯ ВРЕМЕНИ ЗАКРЫТИЯ ЗАДАЧ В JIRA  
THE INFLUENCE OF SOFTWARE DEVELOPMENT  
CHARACTERISTICS ON THE ACCURACY OF MODELS FOR  
PREDICTING ISSUE RESOLUTION TIME IN**



**Федоров Дмитрий Владимирович**, аспирант кафедры индустриального программирования ИПТИП, Федеральное государственное бюджетное образовательное учреждение высшего образования «МИРЭА – Российский технологический университет», Москва

**Юдин Александр Викторович**, д.э.н., к.ф.-м. н., заведующий кафедрой индустриального программирования ИПТИП, Федеральное государственное бюджетное образовательное учреждение высшего образования «МИРЭА – Российский технологический университет», Москва

**Fedorov Dmitry Vladimirovich**, Postgraduate Student of the Department of Industrial Programming at IPTIP, Federal State Budgetary Educational Institution of Higher Education "MIREA – Russian Technological University", Moscow

**Yudin Aleksandr Viktorovich**, Doctor of Economics, Candidate of Physics and Mathematics, Head of the Department of Industrial Programming at the Institute of Industrial Programming, Federal State Budgetary Educational Institution of Higher Education "MIREA – Russian Technological University", Moscow

**Аннотация.** В данной работе исследуется влияние характеристик процесса разработки программного обеспечения на предсказательную способность моделей машинного обучения при прогнозировании времени выполнения задач. Исследование выполнено на открытых обезличенных датасетах проектов Hyperledger, JFrog и Mojang, сформированных на основе данных систем управления задачами разработки. Для прогнозирования использовались модели Random Forest, Gradient Boosting и CatBoost. Применение данных моделей показало устойчивое превосходство моделей машинного обучения над наивным прогнозом, основанным на среднем значении целевой переменной. Средняя абсолютная ошибка была снижена на 37,7-75,5% в зависимости от используемого датасета, при этом наилучший результат достигнут на наборе данных JFrog, где значение MAE уменьшилось с 14948 до 3665 секунд. Анализ значимости признаков показал, что наибольший вклад в качество прогнозирования вносят процессные характеристики, включая количество изменений статуса задачи, число участников выполнения и время до начала работы. Для наиболее значимых процессных признаков значения permutation importance достигали 257-540, существенно превосходя вклад статических характеристик, таких как тип и приоритет задачи. Рассмотренные датасеты обладают различной степенью формализации процессов разработки. Для высокоструктурированных записей JFrog коэффициент детерминации составил 0,76, в то время как в Mojang он не превышает значения в 0,32. Такая вариативность указывает на прямую зависимость точности прогнозов и объяснительную способность ML-моделей от полноты логирования событий цикла задач в системе их учета. Выявлены наиболее значимые характеристики и их принципиальные особенности по отношению к остальным.

**Abstract.** This paper investigates the influence of software development process characteristics on the predictive performance of machine learning models for issue resolution time estimation. The study is based on anonymized open datasets from

the Hyperledger, JFrog, and Mojang projects, derived from issue tracking systems used in software development. Random Forest, Gradient Boosting, and CatBoost models were employed for prediction. The results demonstrate a consistent superiority of machine learning approaches over a naive baseline prediction based on the mean value of the target variable. Mean Absolute Error (MAE) was reduced by 37.7–75.5% depending on the dataset, with the best result achieved on the JFrog dataset, where MAE decreased from 14,948 to 3,665 seconds. Feature importance analysis revealed that process-related characteristics provide the greatest contribution to prediction quality, including the number of status changes, the number of participants involved in task execution, and the time to first progress. For the most influential process features, permutation importance values reached 257–540, substantially exceeding the contribution of static task attributes such as issue type and priority. The datasets exhibit varying degrees of process formalization. For the highly structured JFrog records, the coefficient of determination reached 0.76, while for Mojang it did not exceed 0.32. This variability indicates a direct relationship between prediction accuracy and the explanatory power of ML models on the one hand, and the completeness of event logging throughout the task lifecycle in the tracking system on the other. The most significant features and their principal distinctions from other attributes are identified and discussed.

**Ключевые слова:** машинное обучение, прогнозирование времени выполнения задач, системы управления задачами, Jira, корпоративная разработка программного обеспечения, Random Forest, CatBoost, значимость признаков

**Keywords:** machine learning, issue resolution time prediction, issue tracking systems, Jira, corporate software development, Random Forest, CatBoost, feature importance

## 1. Введение

Современные процессы разработки программного обеспечения характеризуются высокой динамичностью, распределенной структурой команд и значительной вариативностью сроков выполнения задач. Точное прогнозирование этих сроков остается актуальным в гибких методологиях разработки программного обеспечения, к которым относятся Agile и Scrum. Ошибки в прогнозировании решения задач создает риски для выполнения плана спринта в целом и создает дополнительную нагрузку на другие команды, связанные с работой над проектом.

Для управления и ведения учета выполнения задач регулярно используют систему учета заявок Jira. Ее инструментарий обладает рядом функций автоматизированного заполнения данных, что позволяет создать массив данных о задачах и процессе их выполнения в организации. Тем не менее, эти данные обладают качественными отличиями – так были выделены статические (тип задачи, её приоритет и т.д.) и процессуальные характеристики (число изменений статуса задачи, количество участников, количество повторных открытий, временные интервалы между стадиями выполнения).

Целью данной работы является установить, какие характеристики разработки, которые могут быть зафиксированы в системах учета заявок, вносят наибольший вклад в прогнозирование времени закрытия задач для ансамблевых моделей машинного обучения.

Модели машинного обучения активно используют в случаях, когда необходимо выявить сложные нелинейные зависимости. Актуальные ансамблевые алгоритмы, такие как Random Forest, Gradient Boosting и CatBoost представляют собой самые распространенные методы для обработки табличных данных. Именно они будут использованы для обработки данных, извлеченных из Jira.

После формирования моделей для создания прогнозов, будут выделены наиболее значимые признаки, обладающие наибольшей permutation importance для описания наиболее влиятельных факторов и оценки вклада процессуальных и статических характеристик.

На основе сравнения моделей и анализа значимости признаков выявляются ключевые факторы, определяющие точность прогнозирования времени закрытия задач, а также устойчивость моделей в различных проектных контекстах.

Дальнейшие разделы посвящены описанию данных и признаков, методам построения моделей машинного обучения, результатам экспериментальной оценки и анализу значимости характеристик разработки в контексте прогнозирования времени выполнения задач.

## **2. Краткий обзор исследований и подходов**

Существующие подходы к прогнозированию времени выполнения задач в системах управления разработкой можно разделить на три основные группы: статистические модели, методы машинного обучения на табличных данных и гибридные методы, включающие элементы интерпретации процессных метрик [2, 4]:

- статистические и регрессионные модели, основанные на предположении о линейной зависимости между характеристиками задачи и временем выполнения [8];
- методы машинного обучения на структурированных данных систем управления задачами, включающие ансамблевые алгоритмы и градиентный бустинг [12, 13];
- модели анализа процессных метрик и workflow-ориентированные подходы, использующие признаки жизненного цикла задачи и поведенческие характеристики исполнителей [14, 15, 16].

В рамках статистического подхода прогнозирование времени закрытия задач осуществляется через регрессионные зависимости между агрегированными характеристиками проекта и целевой переменной [8].

Линейные модели (например, множественная линейная регрессия) плохо решают задачи, связанные с прогнозированием процессами разработки распределенных команд и Agile-методологий. Современные исследования подтверждают, что с ростом изменчивости процессов, а также при наличии вбросов в Jira-подобных системах, точность таких моделей существенно снижается [4,5].

Ансамблевые модели (Random Forest, Gradient Boosting, CatBoost) обеспечивают на Jira-подобных системах более высокую точность, чем линейные модели, поскольку учитывают перекрестные взаимодействия признаков и нелинейные связи [1,3,9,10]. Согласно данным Li et al. (2022) и Van Oosten et al. (2023), снижение MAE при таком подходе оказывается статистически значимым [12, 13]. Вместе с тем интерпретируемость ансамблей остается ограниченной – рост размерности пространства признаков негативно влияет на объяснительную способность модели.

Процессуальные характеристики жизненного цикла задачи, такие как количество изменений статуса, число исполнителей, обладают большей значимостью для прогнозирования времени работы над задачей, чем статические – к такому выводу приходят Mintgomery, Lüders и Maalej (2022, 2025) в своей серии работ [14,15,16]. Анализ процессуальных характеристик разработки получил развитие в самостоятельном направлении [16].

На данный момент вектор современных исследований направлен в сторону ансамблевых моделей машинного обучения и интеграции процессуальных характеристик. Помимо этого, линейные методы в данном спектре задач становятся значительно менее востребованными [4,12].

При этом остаются открытыми вопросы устойчивости моделей к различным типам проектных данных и доминирующим факторам вариативности времени выполнения задач [9, 14].

### **3. Материалы и методы исследования**

#### **3.1 Извлечение и подготовка данных**

В качестве эмпирической базы исследования использованы открытые наборы данных Jira-подобных систем разработки программного обеспечения. Данные включают информацию о жизненном цикле задач и фиксируют процессные, временные и организационные характеристики выполнения задач. Основные источники данных: проекты Hyperledger, JFrog и Mojang. Объем выборок составляет порядка 8600 – 11500 задач в зависимости от датасета. Единицей наблюдения в рамках исследования является отдельная задача (issue), представленная структурированной записью, описывающей ее жизненный цикл. Целевая переменная определяется как время выполнения задачи, вычисляемое как интервал между моментом создания и моментом окончательного закрытия.

Исходные данные извлекаются из коллекций MongoDB, где каждая коллекция соответствует отдельному проекту. Для каждой задачи используются три основных структурных компонента: идентификатор задачи (key), статические атрибуты (fields) и история изменений (changelog). Ограничение на объем извлекаемых данных позволяет унифицировать структуру входного представления и снизить вычислительную нагрузку на этапах последующей обработки.

Поле key используется как уникальный идентификатор задачи. Поле fields содержит статические характеристики задачи, включая тип задачи (issue\_type), приоритет (priority), исполнителя (assignee), а также временные признаки создания задачи (created\_hour, created\_weekday). Данные признаки представляют контекст постановки задачи и фиксируются на момент ее инициализации в системе. История изменений changelog используется для

формирования процессных характеристик выполнения задач. На ее основе вычисляются метрики количества участников процесса, числа изменений статуса и числа повторных открытий. Дополнительно формируется временной показатель продолжительности выполнения задачи.

На основе исходных и процессных характеристик формируются производные признаки, отражающие нагрузку и сложность выполнения задач. Для оценки загрузки исполнителя используется показатель *assignee\_load*, определяемый как число задач, приходящихся на конкретного исполнителя. На основе полученных характеристик были сформированы следующие производные:

1. Для учета загрузки исполнителей вводится признак *assignee\_load*

$$assignee\_load(a) = \sum_{i=1}^N I(assignee_i = a)$$

2. Для определения общей сложности жизненного цикла задачи используется составной признак *complexity\_proxy*

*complexity\_proxy*

$$= num\_status\_changes + num\_reopens + num\_participants$$

В таблице 1 представлен экземпляр полученного датасета.

Таблица. 1 - Экземпляр датасета JFrog.

issue_type	Improvement
priority	Medium
assignee	<< author_displayName 20e9e0e3-2273-4f47-b7f7-e1eee75dca4c >>
num_participants	2
num_status_changes	2
num_reopens	0

time_to_first_progress	188.6775
time_to_resolution	362.13944444444445
created_hour	11
created_weekday	3
assignee_load	5
complexity_proxy	3

### 3.2. Очистка данных

Очистка данных выполняется на этапе формирования итогового датасета после извлечения признаков из структуры MongoDB-документов. Исходный массив записей преобразуется в pandas DataFrame, после чего применяется фильтрация по качеству целевой переменной. Первый этап очистки связан с удалением записей, в которых отсутствует значение времени разрешения задачи. Такие записи возникают в случаях, когда в истории изменений отсутствует событие Resolved или Closed, либо оно не фиксируется в changelog. Подобные наблюдения исключаются из выборки, поскольку не позволяют вычислить целевую переменную модели.

Второй этап очистки заключается в исключении некорректных значений времени выполнения задач. Удаляются записи, где time\_to\_resolution принимает нулевые или отрицательные значения.

Нормализация в данном случае не вводилась из-за совместимости с используемыми моделями машинного обучения и особенностей постановки задачи регрессии на табличных данных. Используемые алгоритмы (Random Forest, Gradient Boosting и CatBoost) оперируют разбиением признакового пространства и не требуют масштабирования входных переменных, так как не основаны на вычислении расстояний или градиентов в исходном признаковом масштабе.

### 3.3. Используемые методы

В качестве основного метода использовался Random Forest. Алгоритм основан на построении множества деревьев решений, обучаемых на различных подвыборках данных с последующим усреднением результатов. Такая схема снижает дисперсию модели и уменьшает чувствительность к шуму в данных, что критично для Jira-датасетов с высокой вариативностью временных характеристик задач.

Дополнительно применялся Gradient Boosting. Метод реализует последовательное построение ансамбля моделей, где каждая следующая модель корректирует ошибки предыдущих. За счет оптимизации функции потерь на каждом шаге достигается более высокая точность аппроксимации сложных зависимостей, однако возрастает чувствительность к настройке гиперпараметров и структуре данных.

CatBoost был выбран в качестве третьего метода. Он основан на той же логике, что и Gradient Boosting, однако обладает важным отличием. CatBoost оперирует категориальными признаками без их преобразования в числа. Благодаря этому снижается потеря значимой нечисловой информации и повышается стабильность на проектных данных со смешанными типами признаков.

## 4. Результаты исследования

Модели машинного обучения обучались и тестировались на наборах данных из трех Jira-систем: Hyperledger, JFrog и Mojang. Для анализа качества результатов использовались метрики MAE и  $R^2$ . Его результаты представлены на рисунках 1-3. Оценка значимости характеристик разработки произведена благодаря permutation importance и отображена на рисунках 4-6.

Эти датасеты были выбраны, поскольку обладают различной структурой записей. Это отражается в глубине и детализации логирования задач, особенно в статических признаках, таких как issue\_type и priority.

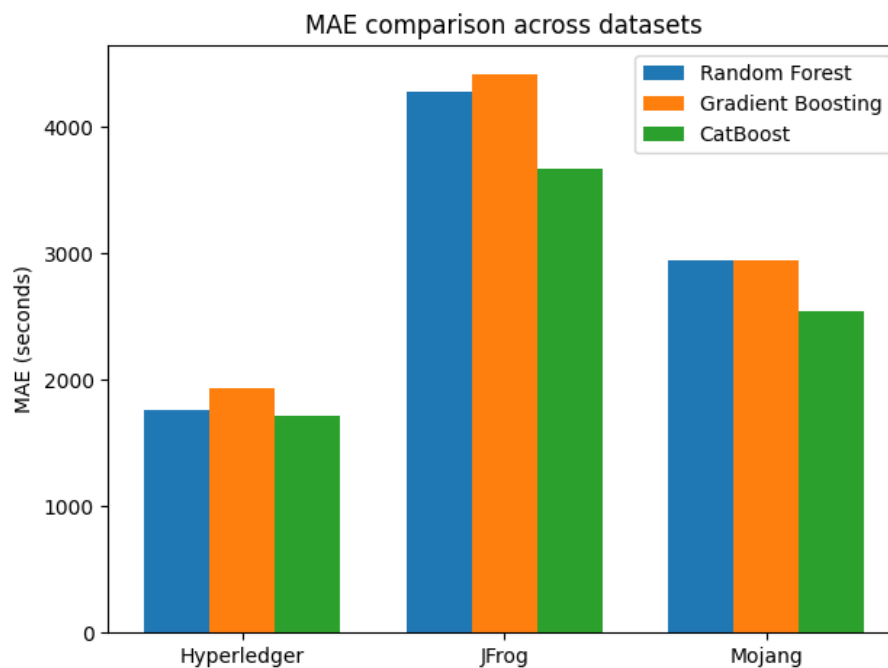


Рисунок 1 - Распределение значений MAE в датасетах Hyperledger, JFrog и Mojang.

На рисунке 1 представлено сравнение моделей Random Forest, Gradient Boosting и CatBoost по метрике средней абсолютной ошибки. Во всех исследуемых датасетах наилучшими показателями обладает модель CatBoost, обеспечившей минимальные ошибки прогнозирования. Так, по сравнению со вторым местом по MAE, CatBoost превзошел их на 2,6% (Hyperledger), 14,2% (Hyperledger), 13,4% (Mojang).

Объясняющая способность моделей оценивалась через коэффициент детерминации  $R^2$ . В условиях шумных Jira-данных значения  $R^2$  составляли 0,5-0,7 на рассматриваемых датасетах (рис. 2) за исключением Mojang. Данные значения можно трактовать как умеренно высокую объяснимость.

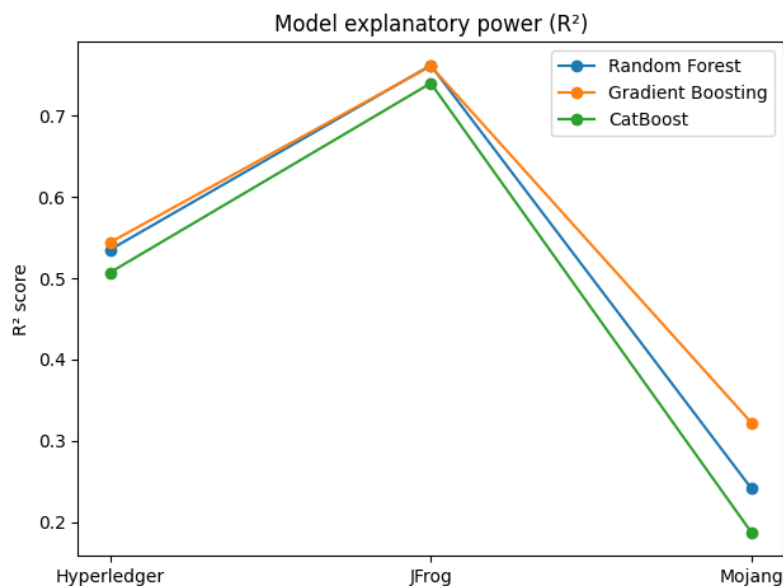


Рисунок 2 - Значения  $R^2$  на датасетах Hyperledger, JFrog и Mojang

Наибольшая объяснительная способность зафиксирована на датасете JFrog. На нем  $R^2 = 0,761$  на модели Random Forest, что более, чем вдвое превосходит показатели Mojang, где лучший  $R^2 = 0,322$ . Датасет Hyperledger показал средние результаты, где  $R^2 = 0,544$ . Наилучшая с точки зрения объясняющей способности модель во всех датасетах была одинакова – Gradient Boosting, кроме датасета JFrog, где Random Forest показал преимущество в  $\Delta R^2 = 0,001$ .

Исходная выборка для каждого репозитория составляла 30000 задач. После этапов подготовки и очистки данных, описанных ранее количество оставшихся задач в каждом датасете распределилось неравномерно. Это подтверждает изначальное утверждение о различном уровне и качестве логирования событий в выбранных организациях. Таким образом, выборка сократилась до следующих размеров:

- Hyperledger – 11500 задач
- JFrog – 9000 задач
- Mojang – 8600 задач

Эта разница объясняется структурой данных: чем последовательнее и детальнее в репозитории фиксируются статусные переходы и события, тем выше предсказуемость времени закрытия задач. Так, в датасете Mojang поле `issue_type` было сформировано из позиций «Bug» в подавляющем большинстве случаев, что сделало этот признак неинформативным.

Сравнение моделей с наивным прогнозом проводилось по формуле:

$$Relative\ improvement = \frac{MAE_{baseline} - MAE_{model}}{MAE_{baseline}} \times 100\%$$

где  $MAE_{baseline}$  – ошибка прогноза по среднему значению целевой переменной (наивный прогноз),  $MAE_{model}$  – ошибка прогноза сравниваемой модели.

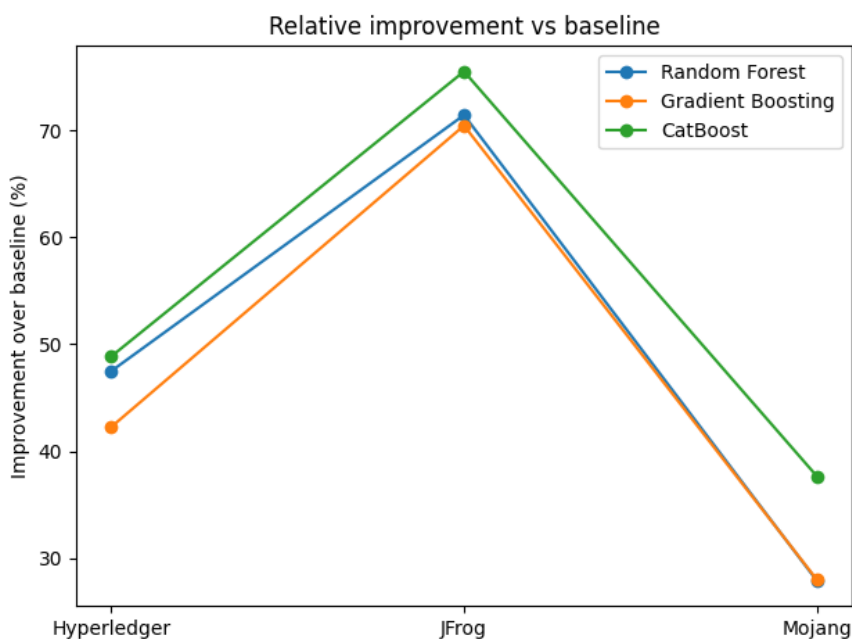


Рисунок 3 - Относительное улучшение прогноза по MAE для всех моделей и датасетов по отношению к наивному прогнозу

Средняя MAE наивного прогноза на предложенных датасетах составляла 3354 секунд в датасете HyperLedger, 14948 секунд в JFrog и 4089 секунд в датасете Mojang. Как видно из рис. 1, была протестирована каждая описанная модель. MAE относительно этого прогноза значительно сократилась на 48.9% у Hyperledger, 75,5% у JFrog, 37,7% у Mojang соответственно. Наибольший положительный эффект в оценке времени закрытия задач

показал датасет JFrog, значимость признаков в оценке которого будет рассмотрена отдельно в пункте «анализ значимости признаков».

Устойчивость моделей проверялась с помощью кросс-валидации. Она применялась для метода Random Forest для датасета JFrog и Gradient Boosting для остальных, показавших наибольшие значения в объясняющей способности. Разброс метрик остается достаточно низким, чтобы заверять о статистической значимости полученных результатов:

– Hyperledger: MAE =  $1958 \pm 25$  с,  $R^2 = 0,54 \pm 0,017$

– JFrog: MAE =  $4353 \pm 69$  с,  $R^2 = 0,79 \pm 0,017$

– Mojang: MAE =  $2957 \pm 85$  с,  $R^2 = 0,32 \pm 0,038$

Качество прогнозирования определяется полнотой и согласованностью процессуальных характеристик в исходных данных. На структурированных репозиториях JFrog модели работают стабильнее и показывают высокий  $R^2$ , в то время как на слабоструктурированных датасетах Mojang объясняющая способность моделей значительно падает, а вместе с ней растет MAE.

Снижение MAE относительно наивного прогноза зафиксировано на всех трех датасетах, что подтверждает выдвинутую идею о том, что учет процессуальных и статических характеристик устойчиво повышает предсказательные способности ML-моделей для времени закрытия задач в Jira-подобных системах. Это повышение варьируется в зависимости от структуры и полноты метрик, фиксируемых в Jira.

## **5. Анализ значимости признаков**

В каждом из рассмотренных датасетов и применяемой ML-модели прослеживаются наиболее значимые признаки. Их состав обладает качественно схожими свойствами – одни и те же процессуальные характеристики зачастую попадают в этот список. В рамках экспериментальной части из полного набора обученных конфигураций были выделены только те модели, которые демонстрировали максимальные значения  $R^2$  на соответствующих датасетах после процедуры валидации.

Такой подход исключает из дальнейшего рассмотрения менее информативные и заведомо слабые варианты, не вносящие дополнительной интерпретативной ценности в анализ.

Процессные характеристики формируют основную информационную нагрузку моделей. К ним относятся количество изменений статуса, число участников выполнения задачи, число переоткрытий и временные интервалы до начала активности. Эти признаки описывают не состояние задачи, а динамику взаимодействий вокруг нее. В терминах модели они выступают как прокси факторы реальной сложности координации работ. Чем больше переходов состояний и участников вовлечено в процесс, тем выше неопределенность траектории выполнения и тем шире распределение целевой переменной.

На всех трех наборах данных сохраняется повторяющийся эффект: метрики, связанные с динамикой статусов и вовлечением участников, стабильно занимают верхние позиции по permutation importance. В Hyperledger доминирующий вклад распределен между числом участников, временем до первого прогресса и исполнителем. В JFrog наблюдается выраженная концентрация влияния в одной переменной, связанной с количеством изменений статуса. В Mojang распределение значимости признаков более равномерно, что объясняется низкой предсказательной способностью всех ML-моделей и высокой MAE в данном датасете.

Для оценки значимости признаков используется перестановочная важность, permutation importance (далее PE).

$$PE = \frac{1}{k} \sum_{i=1}^k (MAE_{shuffled,j,r} - MAE_{model})$$

где  $k = 10$ , число повторений,  $MAE_{model}$  – ошибка на тестовых данных,  $MAE_{shuffled,j,r}$  – ошибка после перемешивания признака  $j$  в повторении  $r$ .

В датасете Hyperledger (рис. 4) ярко выражены два доминирующих признака – исполнитель задачи, где  $PE = 496,7$  с. и число участников

процесса (PE = 470,87 с.). Значительно меньшим влиянием обладает характеристика время до первого прогресса, то есть изменения статуса задачи (PE = 278,9 с). То есть Hyperledger основными предсказательными характеристиками являются те, что относятся к составу работающей над задачей команды, а тип задачи и её приоритет вносят минимальный вклад в качество прогнозирования.

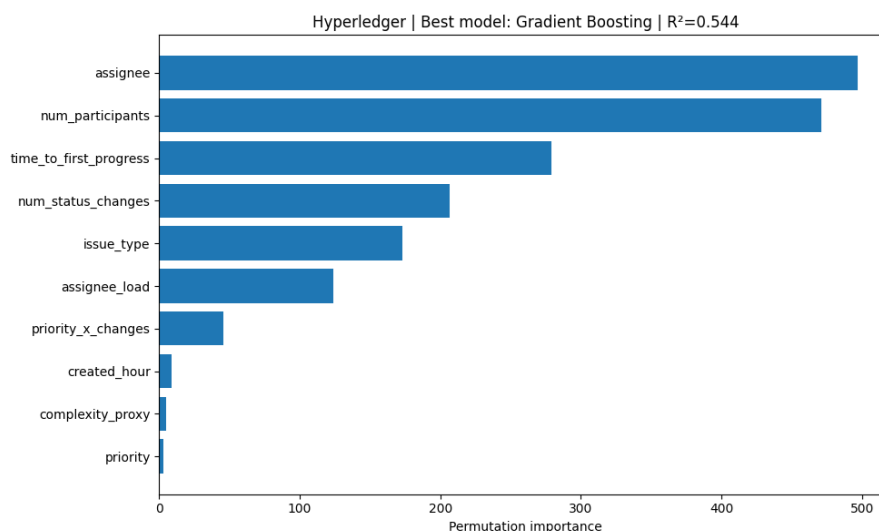


Рисунок 4 - Значимость характеристик разработки в датасете Hyperledger

Разница между датасетами отражает разный уровень формализации процессов разработки в соответствующей организации. В JFrog изменения статуса задач жестко привязаны к этапам работы над задачей, отсюда возникает доминирование признака num\_status\_changes (PE = 9482,49 с). Это значение сохраняется и в Random Forest (PE=9432,26 с). Данный параметр более, чем в 4 раза превышает значимость остальных характеристик.

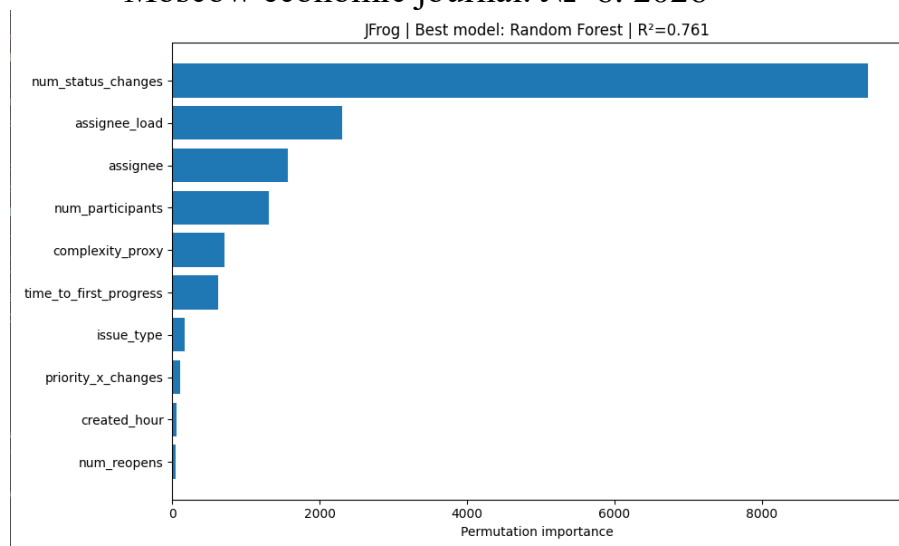


Рисунок 5 - Значимость характеристик разработки в датасете JFrog

Второй по значимости параметр – assignee\_load. Эта производная характеристика, рассчитываемая из количества задач, выделенных на одного сотрудника, обладает достаточно высокой значимостью (PE = 2299 с). Статические характеристики, такие как час создания, день создания и тип задачи находятся в нижней части рейтинга и практически не влияют на прогноз.

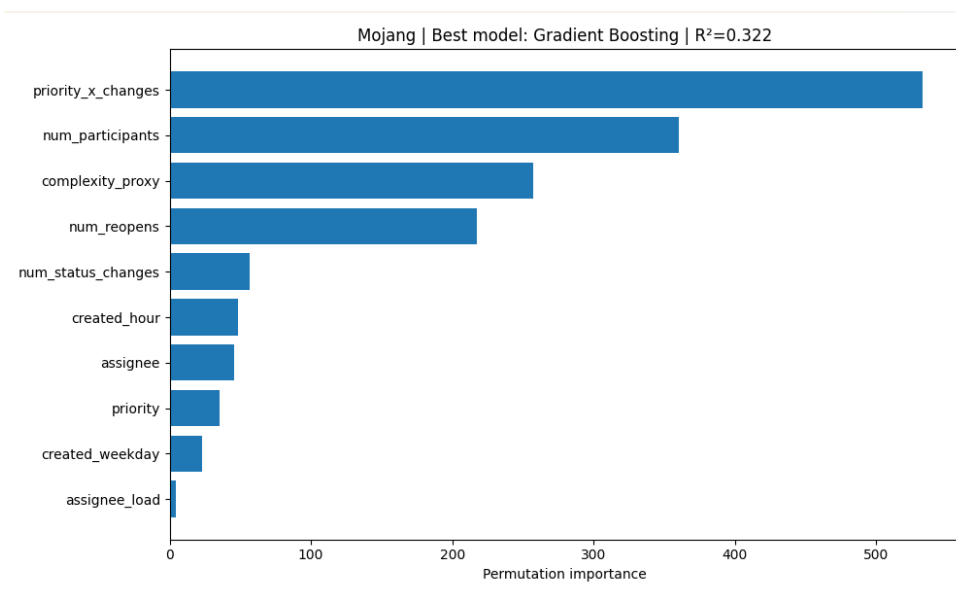


Рисунок 6 - Влияние характеристик разработки по permutation importance в датасете Mojang

Mojang (рисунок 6) распределение важности характеристик разработки более широкое. Но даже в этом случае первые 5 характеристик относятся к процессуальным – `priority_x_changes` (PE= 553,01 с), `num_participants` (PE = 360,68 с), `complexity_proху` (PE = 257, 61 с), `num_reopens` (217,11 с). Остальные имеют значение PE меньше 50. Производные признаки `complexity_proху`, `assignee_load` частично компенсируют низкое качество логирования данных (в данном случае `issue_type` не учитывается поскольку все тикеты в данном датасете имеют одно и тоже текстовое значение «Bug»).

## 6. Обсуждение результатов

Отличия между датасетами подтверждают: качество моделей и структура значимости признаков определяется степенью формализации процессов разработки в профильной организации. Тем не менее, во всех из приведенных примеров процессуальные характеристики обладают более высокой значимостью по сравнению с статическими. Gradient Boosting и CatBoost имеют более низкую MAE, чем Random Forest.  $R^2$  значительно зависит от представленных данных: от 0,32 для Mojang до 0,76 от JFrog. Доказано, что разница обусловлена уровнем формализации процессов и полнотой логирования в Jira-системах.

Структура важности характеристик варьируется: В JFrog доминирует `num_status_changes`, в Hyperledger и Mojang значимость распределена более равномерно. Следовательно, описать универсальную модель, подходящую всем организациям, не представляется возможным. Значимость конкретных признаков зависит от контекста проекта, фиксируемого в Jira-подобных системах.

Тем не менее, прогнозирование сроков выполнения задач в Jira рационально описывать через учет процессуальных характеристик, описывающих жизненный цикл задачи. Использование статических характеристик уместно для создания производных параметров в случаях,

когда предполагается, что в организации изначально существует достаточно низкое качество логирования событий.

## 7. Заключение

В трех датасетах Jira по проектам организаций Hyperledger, JFrog и Mojang были спроектированы и протестированы модели для прогнозирования времени закрытия задач. Были использованы такие ML-модели, как Random Forest, Gradient Boosting и CatBoost. Результаты каждой модели показали значительное улучшение по сравнению с наивным прогнозом по MAE. Лучший результат показала модель CatBoost на датасете JFrog, достигнув значений  $R^2 = 0,76$  и снижением MAE на 75,5%. CatBoost показал наилучшие показатели по параметру снижения MAE на всех рассматриваемых данных. Gradient Boosting – максимальный  $R^2$  на двух из трех датасетов с крайне малым отставанием когда занял второе место.

Анализ значимости признаков выявил устойчивый паттерн. Процессуальные характеристики выполнения задачи (число изменений статуса задачи, количество участников, работающих над задачей, загрузка исполнителя, время до первого изменения статуса) обладают значительно большей значимостью по сравнению со статическими характеристиками, таких как тип задачи, её приоритет, день и время создания. В хорошо структурированных данных датасета JFrog доминирует один признак – num\_status\_changes. В менее формализованных датасетах значимость среди процессуальных характеристик распределена более равномерно, при том сохраняя значимое различие между ранее описанными двумя типами характеристик.

Структура значимости внутри процессуальных и статическими характеристик зависит от уровня формализации заполнения данных в Jira-подобных системах. Выбор алгоритма для их обработки также имеет значение, однако достичь высокой объяснительной способности модели и

низкой МАЕ представляется возможным исключительно в случае, когда логирование задач производится структурировано и детально.

Данная работа имеет ряд ограничений – были рассмотрены всего 3 организации и их проекты. Качество моделей ограничено полнотой логирования. Внешние факторы, такие как приоритет всего проекта в рамках организации, организационные изменения во время его осуществления не учитывались. Permutation importance не устраняет эффект коррелированных признаков.

### Список источников

1. Шевнина, Ю. С. Метод оценки состояния нелинейной системы на основе логического анализа данных / Ю. С. Шевнина // Известия высших учебных заведений. Электроника. – 2022. – Т. 27, № 3. – С. 407-415. – DOI 10.24151/1561-5405-2022-27-3-407-415. – EDN BZRDEB.
2. Бевзенко, С. А. Применение искусственного интеллекта и машинного обучения в разработке программного обеспечения / С. А. Бевзенко // Инновации и инвестиции. – 2023. – № 8. – С. 187-191. – EDN ODNEIS.
3. Салтанаева, Е. А. Сравнение традиционных методов машинного обучения и глубокого обучения / Е. А. Салтанаева, А. А. Шакиров, А. Р. Гимаева // Научно-технический вестник Поволжья. – 2023. – № 12. – С. 379-381. – EDN EQNUHN.
4. Леохин, Ю. Л. Методы машинного обучения в прикладных задачах прогнозирования динамично изменяющихся данных / Ю. Л. Леохин, С. С. Дымкова, Т. Д. Фатхулин // Т-Сomm: Телекоммуникации и транспорт. – 2025. – Т. 19, № 8. – С. 49-63. – DOI 10.36724/2072-8735-2025-19-8-49-63. – EDN ULVCHG.
5. Гудков, А. А. Прогнозирование эффективности проектной деятельности на основе интеграции подходов бизнес-аналитики и машинного обучения / А. А. Гудков // Вестник Волжского университета им. В.Н. Татищева. – 2024. – Т.

2, № 1(53). – С. 27-36. – DOI 10.51965/2076-7919\_2024\_2\_1\_27. – EDN QYXZHC.

6. Чистякова, К. А. Практические методы управления реализацией инновационных проектов на основе использования программного обеспечения “Jira” / К. А. Чистякова, В. В. Юдин // Наука и искусство управления / Вестник Института экономики, управления и права Российского государственного гуманитарного университета. – 2023. – № 1. – С. 80-93. – DOI 10.28995/2782-2222-2023-1-80-93. – EDN KXHBSV.

7. Коротких, А. В. Методы автоматизированной оценки трудоёмкости задач разработки программного обеспечения / А. В. Коротких, И. В. Потапов // ИТ. Наука. креатив : Материалы I Международного форума: в 5-ти томах, Омск, 14–16 мая 2024 года. – Москва: Общество с ограниченной ответственностью "Издательско-книготорговый центр "Колос-с", 2024. – С. 213-218. – EDN YPZFCG.

8. Нестеров, Ю. Г. Подход к применению машинного обучения в прогнозировании загрузки виртуальных вычислительных систем / Ю. Г. Нестеров, А. П. Калистратов, Г. И. Афанасьев // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. – 2019. – № 11-2. – С. 73-76. – EDN PИИИИИ.

9. Телегин, В. А. Использование методов машинного обучения для создания алгоритма адаптивной оценки времени выполнения проектных задач / В. А. Телегин // Инновационные научные исследования. – 2023. – № 6-3(30). – С. 146-161. – DOI 10.5281/zenodo.8128520. – EDN PXCIGW.

10. Петрунько, А. О. Применение методов машинного обучения при управлении инновационными проектами / А. О. Петрунько, М. Ф. Иванов // Естественно-гуманитарные исследования. – 2025. – № 3(59). – С. 909-915. – EDN JUENGB.

11. Сапунов, А. В. Использование цифровых технологий в принятии управленческих решений / А. В. Сапунов, Т. А. Сапунова // Вестник Академии знаний. – 2023. – № 1(54). – С. 235-238. – EDN FQOQNR.
12. Li, Y. Identifying self-admitted technical debt in issue tracking systems using machine learning / Y. Li, M. Soliman, P. Avgeriou // Empirical Software Engineering. – 2022. – Vol. 27, No. 6. – P. 1-37. – DOI 10.1007/s10664-022-10128-3. – EDN JTDVOO.
13. Van Oosten W., Rasiman R., Dalpiaz F., Hurkmans T. On the effectiveness of automated tracing from model changes to project issues // Information and Software Technology. 2023. Vol. 161. Article 107226. DOI: 10.1016/j.infsof.2023.107226.
14. Montgomery L., Lüders C., Maalej W. An alternative issue tracking dataset of public Jira repositories // Proceedings of the 19th International Conference on Mining Software Repositories. – 2022. – С. 73-77.
15. Lüders C. M., Pietz T., Maalej W. Automated detection of typed links in issue trackers // 2022 IEEE 30th International Requirements Engineering Conference (RE). – IEEE, 2022. – С. 26-38.
16. Montgomery L., Lüders C., Maalej W. Mining issue trackers: Concepts and techniques // Handbook on Natural Language Processing for Requirements Engineering. – Cham : Springer Nature Switzerland, 2025. – С. 309-336.

### References

1. Shevnina, Yu. S. Metod ocenki sostoyaniya nelinejnoj sistemy` na osnove logicheskogo analiza danny`x / Yu. S. Shevnina // Izvestiya vy`sshix uchebny`x zavedenij. E`lektronika. – 2022. – T. 27, № 3. – S. 407-415. – DOI 10.24151/1561-5405-2022-27-3-407-415. – EDN BZRDEB.
2. Bevzenko, S. A. Primenenie iskusstvennogo intellekta i mashinnogo obucheniya v razrabotke programmogo obespecheniya / S. A. Bevzenko // Innovacii i investicii. – 2023. – № 8. – S. 187-191. – EDN ODNEIS.

3. Saltanaeva, E. A. Sravnenie tradicionny`x metodov mashinnogo obucheniya i glubokogo obucheniya / E. A. Saltanaeva, A. A. Shakirov, A. R. Gimaeva // Nauchno-texnicheskij vestnik Povolzh`ya. – 2023. – № 12. – S. 379-381. – EDN EQNUHN.
4. Leoxin, Yu. L. Metody` mashinnogo obucheniya v prikladny`x zadachax prognozirovaniya dinamichno izmenyayushhixsya dannyx / Yu. L. Leoxin, S. S. Dy`mkova, T. D. Fatxulin // T-Comm: Telekommunikacii i transport. – 2025. – Т. 19, № 8. – S. 49-63. – DOI 10.36724/2072-8735-2025-19-8-49-63. – EDN ULVCHG.
5. Gudkov, A. A. Prognozirovaniye e`ffektivnosti proektnoj deyatel`nosti na osnove integracii podxodov biznes-analitiki i mashinnogo obucheniya / A. A. Gudkov // Vestnik Volzhskogo universiteta im. V.N. Tatishheva. – 2024. – Т. 2, № 1(53). – S. 27-36. – DOI 10.51965/2076-7919\_2024\_2\_1\_27. – EDN QYXZHC.
6. Chistyakova, K. A. Prakticheskie metody` upravleniya realizaciej innovacionny`x proektov na osnove ispol`zovaniya programmnoho obespecheniya “Jira” / K. A. Chistyakova, V. V. Yudin // Nauka i iskusstvo upravleniya / Vestnik Instituta e`konomiki, upravleniya i prava Rossijskogo gosudarstvennogo gumanitarnogo universiteta. – 2023. – № 1. – S. 80-93. – DOI 10.28995/2782-2222-2023-1-80-93. – EDN KXHBSV.
7. Korotkix, A. V. Metody` avtomatizirovannoj ocenki trudoyomkosti zadach razrabotki programmnoho obespecheniya / A. V. Korotkix, I. V. Potapov // IT. Nauka. kreativ : Materialy` I Mezhdunarodnogo foruma: v 5-ti tomah, Omsk, 14–16 maya 2024 goda. – Moskva: Obshhestvo s ogranichennoj otvetstvennost`yu Izdatel`sko-knigotorgovy`j centr Kolos-s, 2024. – S. 213-218. – EDN YPZFCG.
8. Nesterov, Yu. G. Podxod k primeneniyu mashinnogo obucheniya v prognozirovanii zagruzki virtual`ny`x vy`chislitel`ny`x sistem / Yu. G. Nesterov, A. P. Kalistratov, G. I. Afanas`ev // Sovremennaya nauka: aktual`ny`e problemy` teorii i praktiki. Seriya: Estestvenny`e i texnicheskie nauki. – 2019. – № 11-2. – S. 73-76. – EDN PIIMIH.

9. Telegin, V. A. Ispol`zovanie metodov mashinnogo obucheniya dlya sozdaniya algoritma adaptivnoj ocenki vremeni vy`polneniya proektny`x zadach / V. A. Telegin // Innovacionny`e nauchny`e issledovaniya. – 2023. – № 6-3(30). – S. 146-161. – DOI 10.5281/zenodo.8128520. – EDN PXCIGW.
10. Petrun`ko, A. O. Primenenie metodov mashinnogo obucheniya pri upravlenii innovacionny`mi proektami / A. O. Petrun`ko, M. F. Ivanov // Estestvenno-gumanitarny`e issledovaniya. – 2025. – № 3(59). – S. 909-915. – EDN JUENGB.
11. Sapunov, A. V. Ispol`zovanie cifrovyy`x tehnologij v prinyatii upravlencheskix reshenij / A. V. Sapunov, T. A. Sapunova // Vestnik Akademii znaniy. – 2023. – № 1(54). – S. 235-238. – EDN FQOQNR.
12. Li, Y. Identifying self-admitted technical debt in issue tracking systems using machine learning / Y. Li, M. Soliman, P. Avgeriou // Empirical Software Engineering. – 2022. – Vol. 27, No. 6. – P. 1-37. – DOI 10.1007/s10664-022-10128-3. – EDN JTDVOO.
13. Van Oosten W., Rasiman R., Dalpiaz F., Hurkmans T. On the effectiveness of automated tracing from model changes to project issues // Information and Software Technology. 2023. Vol. 161. Article 107226. DOI: 10.1016/j.infsof.2023.107226.
14. Montgomery L., Lüders C., Maalej W. An alternative issue tracking dataset of public Jira repositories //Proceedings of the 19th International Conference on Mining Software Repositories. – 2022. – S. 73-77.
15. Lüders C. M., Pietz T., Maalej W. Automated detection of typed links in issue trackers //2022 IEEE 30th International Requirements Engineering Conference (RE). – IEEE, 2022. – S. 26-38.
16. Montgomery L., Lüders C., Maalej W. Mining issue trackers: Concepts and techniques //Handbook on Natural Language Processing for Requirements Engineering. – Cham : Springer Nature Switzerland, 2025. – S. 309-336.

© Федоров Д.В., Юдин А.В., 2026. Московский экономический журнал, 2026,

№ 6.