

# ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ И ВОЗМОЖНОСТЕЙ ПРИМЕНЕНИЯ ПРЕДИКТИВНОЙ АНАЛИТИКИ В РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Микитюк Т.Е. (Университет ИТМО)

Научный руководитель – кандидат технических наук Коцюба И.Ю.  
(Университет ИТМО)

**Введение.** Использование предиктивной аналитики способно радикально изменить подход к разработке программного обеспечения. С помощью статистических методов исторические данные, накопленные в ходе выполнения предыдущих задач, можно использовать для прогнозирования того, как сегодняшние действия и события повлияют на результаты в будущем.

Подобный переход к более научному подходу к принятию решений уже набирает обороты среди профессионалов в области программного обеспечения. Предиктивная аналитика позволяет командам разработки оптимизировать постановку задач и непрерывную доставку программного обеспечения для более точного прогнозирования их возможностей и скорости, что приводит к лучшему использованию ресурсов и, в конечном итоге, к более качественному программному обеспечению.

**Основная часть.** В основе предиктивной аналитики лежит использование математических алгоритмов для выявления и объяснения закономерностей в данных для составления прогнозов будущего поведения этих данных. В предиктивной аналитике используется несколько алгоритмов, в том числе:

1. Алгоритмы регрессии. Существует множество различных типов алгоритмов регрессии, все они предназначены для оценки взаимосвязи между переменными. Они варьируются от простых линейных регрессионных моделей до нелинейной регрессии.
2. Анализ временных рядов. Позволяет найти взаимосвязи между данными, собранными за определенный промежуток времени, чтобы предсказать, как они могут повести себя в будущем.
3. Машинное обучение. Алгоритмы машинного обучения изучают входные данные и пытаются найти взаимосвязь с определенным набором данных. По мере обработки большего количества данных алгоритм машинного обучения уточняет взаимосвязь до тех пор, пока не будет получено хорошее соответствие.

Предиктивная аналитика может быть использована для решения проблем в жизненном цикле разработки программного обеспечения. В течение многих лет разработчики получали данные в процессе планирования, создания, тестирования и развертывания своего программного обеспечения. Однако, до недавнего времени эти данные простаивали, пока их не удаляли, чтобы освободить место для новых.

Анализ программного обеспечения может применяться на каждом этапе разработки для лучшего понимания того, что, скорее всего, произойдет. Результаты объединяются для демонстрации постоянно обновляемого сквозного обзора вероятных результатов процесса разработки. Среди основных аспектов, доступных для анализа, можно выделить следующие:

1. Планирование проекта разработки. Применяя предиктивную аналитику при планировании списка задач, становится возможным разработать более точные временные оценки. Объединив данные по нескольким командам, можно построить точный план бэклога для релиза и установить более реалистичные цели доставки.

2. Разработка программного продукта. В процессе создания и тестирования программного обеспечения разработчики и тестировщики генерируют большое количество данных. Методы прогнозирования позволяют повысить эффективность и скорость выполнения работ, одновременно повышая качество выпускаемой продукции.

3. Анализ и операционная деятельность. Методы прогнозирования позволяют понять, как программное обеспечение будет использоваться рядовым или продвинутым пользователем, и помогают снизить вероятность того, что дефекты попадут в производство.

**Выводы.** В ходе исследования были выявлены основные алгоритмы и задачи, решаемые методами предиктивной аналитики и моделирования, которые позволили выявить основные этапы разработки программного обеспечения, на которые может кардинально влиять прогнозирование. Были описаны способы внедрения методов машинного обучения для прогнозирования серьезности дефектов и ошибок программного обеспечения на этапах его разработки и тестирования. Предполагается, что данное исследование будет служить ориентиром для будущих работ и исследований в области прогнозирования дефектов программного обеспечения.

#### **Список использованных источников:**

1. Yanming Yang, Xin Xia, David Lo, Tingting Bi, John Grundy, and Xiaohu Yang. 2022. Predictive Models in Software Engineering: Challenges and Opportunities. *ACM Trans. Softw. Eng. Methodol.* 31, 3, Article 56 (July 2022), 72 pages. <https://doi.org/10.1145/3503509>.

2. Priya Varshini AG, Anitha Kumari K (2020) Predictive analytics approaches for software effort estimation: A review. *Indian Journal of Science and Technology.* 13(21): 20942103. <https://doi.org/10.17485/IJST/v13i21.573>.

3. Federica Sarro. 2018. Predictive analytics for software testing: keynote paper. In *Proceedings of the 11th International Workshop on Search-Based Software Testing (SBST '18)*. Association for Computing Machinery, New York, NY, USA, 1. <https://doi.org/10.1145/3194718.3194730>.

4. T. O. Olaleye, O. T. Arogundade, Sanjay Misra, A. Abayomi-Alli, Utku Kose, "Predictive Analytics and Software Defect Severity: A Systematic Review and Future Directions", *Scientific Programming*, vol. 2023, Article ID 6221388, 18 pages, 2023. <https://doi.org/10.1155/2023/6221388>.