

Министерство образования и науки Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ  
Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра дискретной математики  
и информационных технологий

**Машинное обучение в сфере автоматизации тестирования биржевых  
протоколов**

**АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

Студента 5 курса 521 группы  
направления 09.03.01 «Информатика и вычислительная техника»  
факультета компьютерных наук и информационных технологий  
Смирнова Сергея Николаевича

Научный руководитель

д. э. н., профессор

\_\_\_\_\_

подпись, дата

Л.В. Кальянов

Зав. кафедрой

к. ф.-м.н., доцент

\_\_\_\_\_

подпись, дата

Л.Б. Тяпаев

Саратов 2018

## ВВЕДЕНИЕ

В последние годы машинное обучение с каждым днем занимает всё большее место в нашей жизни, благодаря его огромному спектру применений. Всё больше задач решается с помощью машинного обучения, например подбор контекстной рекламы или анализ пробок и автопилотирование автомобилей.

**Актуальность данной работы** обусловлена востребованностью технологий машинного обучения в сфере автоматизации функционального тестирования технологическими компаниями. Наиболее частый сценарий на сегодняшний день – тестирование систем вручную. Минусы подобного подхода очевидны, к ним стоит отнести человеческий фактор, необходимость каждый раз обучать нового сотрудника, рутинность, высокие временные затраты. Использование технологий машинного обучения позволит облегчить выполнение тестирования, сократить затраты и позволит агрегировать и переиспользовать опыт накопленный сотрудниками в одном месте.

**Целью данной работы** является разработка и внедрение алгоритмов на основе современных технологий машинного обучения, в сферу функционального тестирования биржевых протоколов.

**Задачами данной работы** являются:

- исследовать возможности машинного обучения для применения в сфере автоматизации тестирования;
- провести анализ методов машинного обучения на тестовых данных;
- реализовать обработку и фильтрацию обучающих примеров;
- создать RESTfull веб-сервис предоставляющий прогнозы на основе выбранного метода машинного обучения;
- реализовать графический интерфейс пользователя.

**Работа состоит из** введения, четырех глав, заключения и списка использованных источников.

Первая глава «Машинное обучение в функциональном тестировании» содержит краткий обзор программного продукта Sailfish применяющегося в сфере автоматизации тестирования. Описывает проблему возникшую в сфере автоматизации функционального тестирования, а также пути её решения. Во второй главе «Выбор методов машинного обучения для решения поставленной задачи» проведено сравнение методов машинного обучения, осуществлена оценка работы методов классификации, с выбором оптимального метода для решения поставленной задачи. В третьей главе «Выбор инструментов разработки» описывается выбор языка программирования, архитектуры продукта и используемых библиотек. В четвертой главе «Разработка и интеграция в Sailfish сервиса машинного обучения» описывается процесс разработки RESTfull веб-сервиса с использованием адаптивного бустинга, а также интеграция разработанного сервиса с программным продуктом Sailfish. В заключении приведены основные результаты и выводы по проделанной работе.

В данной работе данные обрабатывались и анализировались при помощи API библиотеки машинного обучения Weka. Weka это очень мощный инструмент машинного обучения и интеллектуального анализа данных, развиваемый мировым научным сообществом, свободно распространяемый под лицензией GNU GPL. Цели проекта, создать современную среду для разработки методов машинного обучения и применения их к реальным данным, сделать методы машинного обучения доступными для повсеместного применения. Предполагается, что с помощью данной среды специалист в прикладной области сможет использовать методы машинного обучения для извлечения полезных знаний непосредственно из данных, возможно, очень большого объёма.

Пользователями Weka являются исследователи в области машинного обучения и прикладных наук. Она также широко используется в учебных целях. Weka содержит средства для предварительной обработки данных, классификации, регрессии, кластеризации, отбора признаков, поиска ассоциативных правил и визуализации. Weka хорошо подходит для разработки новых подходов в машинном обучении. Это, безусловно, очень полезно, при необходимости быстро сравнить несколько алгоритмов на собранных данных. В то же время можно использовать Weka для серьезного интеллектуального анализа данных на выборках умеренного размера.

Для разработчика Weka предоставляет неплохую документацию, а благодаря открытой лицензии всегда можно самому расширить исходный в своих целях. Разработчики могут непосредственно применять алгоритмы к выборкам данных, а также вызывать алгоритмы из программ на языке Java. Weka имеет реализации большинства (по крайней мере, распространенных) алгоритмов машинного обучения, а также позволяет запускать алгоритмы обучения для обработки естественного языка (natural language processing).

Weka позволяет работать с данными в нескольких форматах и из нескольких источников, включая файлы и базы данных. Что позволяет легко

переносить собранные выборки данных между несколькими фреймворками машинного обучения.

Sailfish – это инструмент для автоматизации функционального тестирования. С его помощью можно осуществлять функциональное тестирование систем общающихся с помощью сетевых протоколов, внешних API и других способов взаимодействия.

Одной из отличительных черт данного инструмента является язык Automation Matrix Language (далее в этом документе AML). С его помощью описываются тестовые ситуации, состоящие из последовательности действий взаимосвязанных между собой. Конечный скрипт представляет собой матрицу в формате csv, xls,xlsx, каждая строка которого является отдельной инструкцией.

В ходе выполнения скрипта производится составление HTML отчета с описанием входных параметров инструкций и ответных действий систем с которыми осуществляется взаимодействие. В процессе выполнения скриптов возникают проблемные ситуации. На текущий момент расследование проблемы складывается из следующих шагов:

- Поиск первого провалившегося действия;
- Просмотр альтернативных сообщений;
- Восстановление логики матрицы;
- Поиск места расхождения цепочки планируемых событий и произошедших при выполнении скрипта;
- Анализ ситуации и поиск причины в близлежащих событиях.

Мы видим, что одно из действий провалилось, но пока непонятно почему. Применяя интуицию и человеческий интеллект, попробуем понять в чем же причина. Можно кликать во все 3 Similar Messages, но для опытного пользователя сразу понятно, что надо смотреть именно последний – в нём мы найдём ответ на вопрос, или во всяком случае ключевую подсказку, что же пошло не так. Но это не очевидно для всех пользователей и это нормальная ситуация.

Добавив в Sailfish функциональность запоминания выбора Similar Message (вместе с другими параметрами), то возможно рано или поздно, он научился бы предлагать пользователю куда смотреть и потребность анализа проблемы согласно алгоритму описанному выше отпала бы совсем, либо частично.

Наша задача, – научить машину делать вывод, может ли сообщение являться объяснением причины провала из входных данных, т.е. имеем задачу классификации сообщений отвечающих на вопрос – может ли данное сообщение объяснить причину провала. В данном случае обучение относится к категории обучение с учителем (supervised learning). Возьмем в качестве входных данных набор проваленных действий, для которых известно, какие сообщения из предложенных могут описать причину проблемы. Разделим его на два множества: множество сообщений отмеченных пользователем как вероятное решение проблемы и в противоположность, множество не имеющее требуемых отметок. Далее заставляем машину считать вероятность того, к какому множеству принадлежит новое сообщение, на основании выученных ей корреляций.

Последовательность действий при сборе информации. Пользователь расследует проблему используя данные тестовых ситуаций в HTML отчете, затем отмечает провалившиеся действие кликом по Collect Machine Learning data for this action. После выбора сообщения путем установки отметки Problem explanation у искомого сообщения, пользователь может подтвердить свой выбор кликнув по кнопке Submit data. Процесс сбора обучающих примеров на рисунке 1.

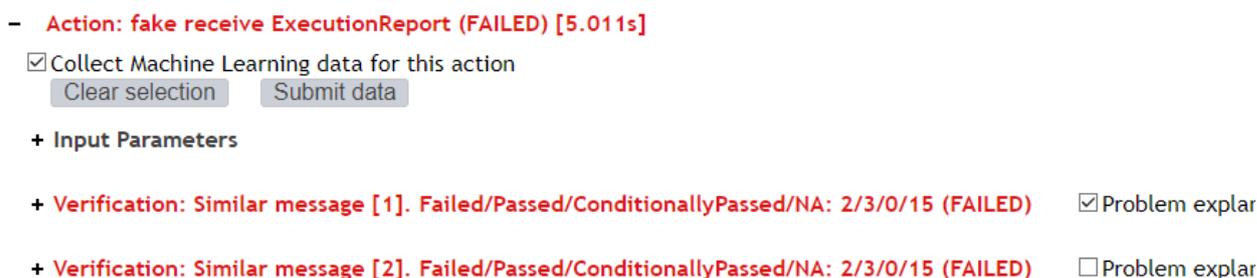


Рисунок 1 – Расследованное провалившееся действие

При просмотре HTML отчетов содержащих проваленные действия в случае доступности установленного и настроенного сервиса машинного обучения автоматически будет произведен AJAX запрос на классификацию сообщений в просматриваемом действии.

В случае получения сообщений классифицированных как возможно объясняющих причину с вероятностью более 50%, будет сформирована подсказка пользователю. Это означает, что стоит обратить внимание в первую очередь на них. Подсказка имеет салатовый цвет фона, яркость которого тем выше, чем выше полученная вероятность. Если представлено более одной подсказки, то они сортируются по убыванию вероятности, Дабы не вводить пользователя в заблуждение, количество отображаемых подсказок было ограничено до 10 шт. Подсказки группируются в таблицу на рисунке 2, содержащую имя сообщения, его человеко читаемое представление и вероятность ответа. Дополнительно подсвечивается блок Similar Messages в котором пользователь производит поиск причины провала.

- **Action: fake receive ExecutionReport (FAILED) [5.011s]**
  - + **Predictions: 2 PCs**
    - Collect Machine Learning data for this action
  - + **Input Parameters**
  - + **Verification: Similar message [1]. Failed/Passed/ConditionallyPassed/NA: 2/3/0/15 (FAILED)**
  - + **Verification: Similar message [2]. Failed/Passed/ConditionallyPassed/NA: 2/3/0/15 (FAILED)**
  - + **Status**

Рисунок 2 – Подсветка Similar messages с приоритетом

## ЗАКЛЮЧЕНИЕ

В ходе выполнения ВКР исследованы возможности применения технологий машинного обучения в сфере автоматизации тестирования биржевых протоколов. Рассмотрен ряд методов машинного обучения предоставляющих функциональность классификации входных данных.

Произведена тренировка рассматриваемых прогностических моделей и сделан выбор наиболее подходящей модели. Реализована дополнительная обработка и фильтрация обучающих примеров для достижения оптимальных результатов.

Был разработан RESTfull веб сервис, позволяющий производить классификацию входных данных с использованием предварительно обученной прогностической модели. Для достижения этой цели были изучены основные возможности библиотеки Weka, обучена и оптимально настроена AdaBoost.M1 модель. Также были изучены принципы архитектуры REST. Изучена технология развертывания сервера приложений Tomcat и его правильная настройка. Реализован графический интерфейс пользователя с использованием технологий HTML5, CSS, JavaScript. Таким образом, все поставленные задачи выполнены.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Weka AdaBoostM1 [Электронный ресурс] // Официальная документация библиотеки машинного обучения Weka по методу классификации AdaBoostM1. [Электронный ресурс] : [сайт]. URL: <http://weka.sourceforge.net/doc.dev/weka/classifiers/meta/AdaBoostM1.html> (дата обращения: 12.05.2018). Загл. с экрана. Яз.: англ.

2 Weka ARFF API [Электронный ресурс] // Официальная документация библиотеки машинного обучения Weka по работе с api наборов данных. [Электронный ресурс] : [сайт]. URL: <http://weka.wikispaces.com/Creating+an+ARFF+file> (дата обращения 14.05.2018). Загл. с экрана. Яз.: англ.

3 MachineLearningMastery [Электронный ресурс] // Boosting and AdaBoost for Machine Learning. [Электронный ресурс] : [сайт]. URL: <https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/> (дата обращения 14.05.2018). Загл. с экрана. Яз.: англ.

4 MachineLearningMastery [Электронный ресурс] // Difference Between Classification and Regression in Machine Learning. [Электронный ресурс] : [сайт]. URL: <https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/> (дата обращения 14.05.2018). Загл. с экрана. Яз.: англ.