

Предсказание времени жизни строчек кода

Студент:
Терехов Арсений

Консультант:
Поваров Никита

Научный руководитель:
Тимофей Брыксин 1/14



Что такое дефект в исходном коде ПО?

- Запись в системе отслеживания ошибок
- Синтезация неправильного кода из корректного
- Время жизни строки кода



Почему именно время жизни строки?

- Строго определенное понятие
- Нужен только git
- Содержит в себе информацию про ошибки и не только

Цель проекта

- Сделать инструмент для привлечения внимания разработчиков к часто-изменяемому коду.

Задачи

- Получить решающую функцию, предсказывающую через сколько дней строку либо изменят, либо удалят

$h: \text{features}(\text{file}, \text{line}, \text{commit}) \rightarrow R$

- Разработать плагин для платформы IntelliJ IDEA, выводящий результаты предсказаний пользователю

Аналоги



Per Norfords: Prediction of code lifetime, 2017, Linköping University.

- Время жизни вычисляется в коммитах
- Предсказывается время жизни для непрерывной последовательности добавленных строк
- **Сложная практическая применимость**
- **Нет инструмента, который бы это предсказывал**

Признаки и сбор датасета

Source Code

Keywords
Location
Depth
Length
...



Current time life
Author experience
Commit count
...

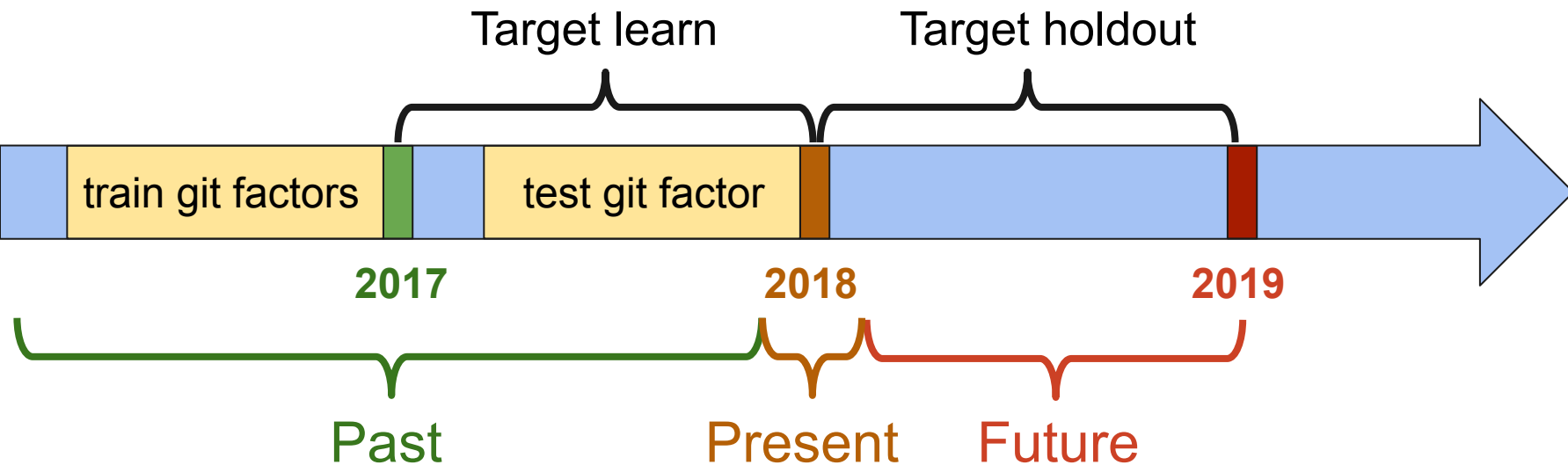


BestPractices
Error Prone
CodeStyle
...

Решающая функция - Random Forest

- Нелинейная функция, позволяет выразить **сложные зависимости**
- Способность эффективно обрабатывать данные с большим числом признаков и классов
- Вычисления во время обучения и предсказаний **можно производить параллельно**
- Не склонен к переобучению

Оценка качества предсказаний



Текущие результаты

Regression

Metric-Model	Test 2017	Holdout 2018
MAE-RF	62.8	65.1
MAE-BASE	80.2	89.4

Classification

Metric-Model	Test 2017	Holdout 2018
AUC-RF	0.76	0.65
AUC-BASE	0.5	0.5

Демонстрация UI

```
1  /**
16 package com.intelli
17
18 import ...
19
30 /**
31  * All extensions c
32  * @link Extension
33  * is another same-
34  * that has "first"
35  *
36  * Extension ID can
37  * to specify also.
38  *
39  * If some anchor i
40  *
41  * @author Alexande
42
43 public class LoadingOrder {
44     @NonNls private s
45     @NonNls private s
46     @NonNls private s
47     @NonNls private s
48     @NonNls private s
49     @NonNls private s
50
51     public static fin
52     public static fin
53     public static fin
54
55     @NonNls private f
56     private final boo
57     private final boo
58     private final Set
59     private final Set
60
61     private LoadingOr
62     myName = "ANY";
63     myFirst = false
64     myLast = false;
65 }
66
```

42 public class LoadingOrder {

43 @NonNls private static final

44

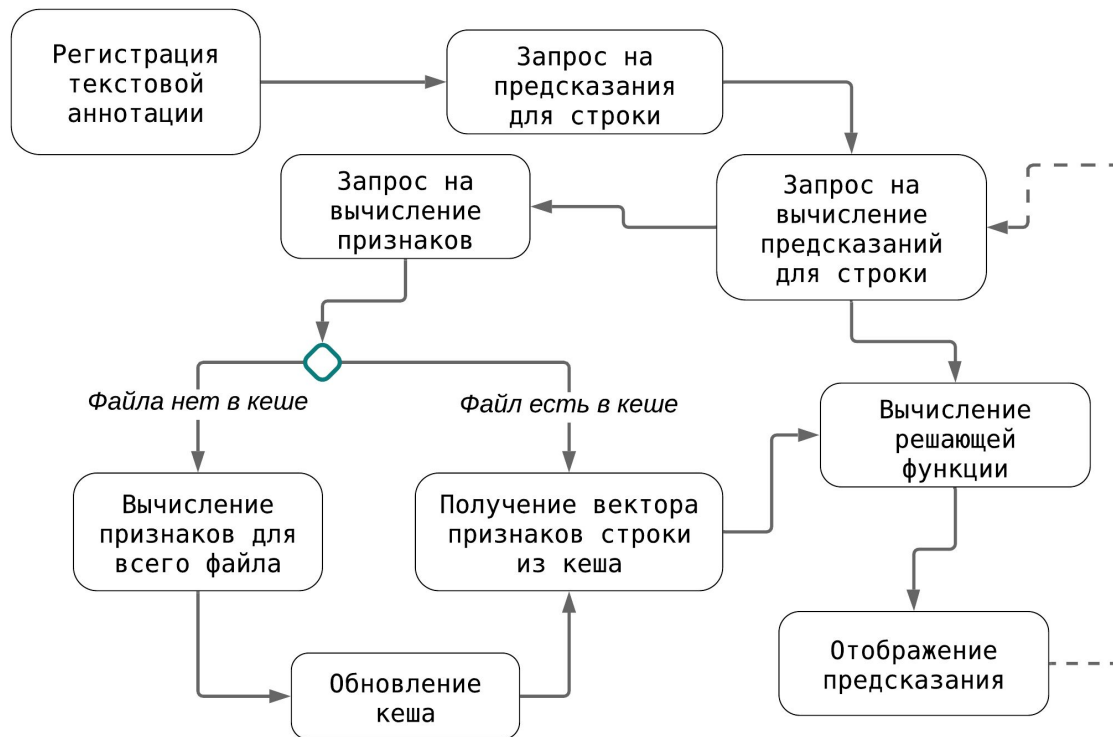
45 3 years, 9 months, 2 weeks, 6 days

46 @NonNls private static final

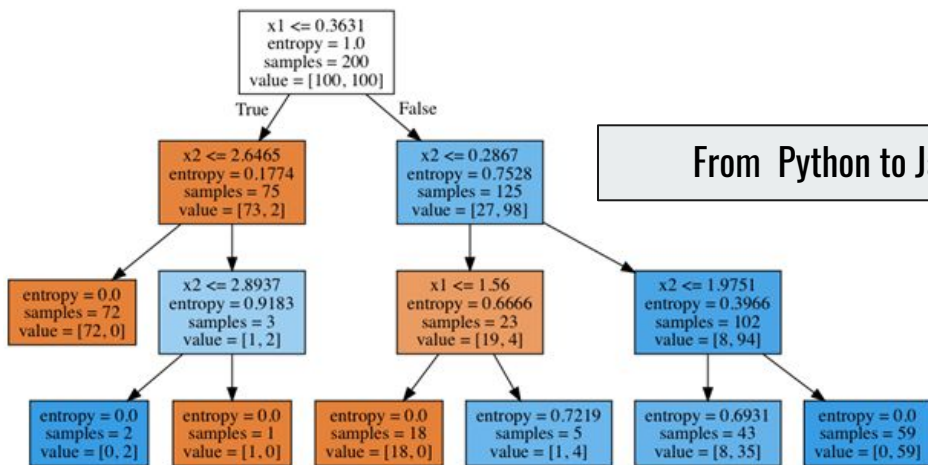
47 @NonNls private static final

48 @NonNls private static final

Диаграмма активностей



Встраивание решающей функции



From Python to Java

```
//...
if (fs[21] <= 1905.5) {
    if (fs[28] <= 4.5) {
        return 0.90752481723334;
    } else {
        return 0.8417089341665657;
    }
} else {
    if (fs[33] <= 707.5) {
        return 0.9136697481129675;
    } else {
        return 0.0;
    }
}
//...
```

Итоги



- Написан код автоматической сборки датасета и собраны несколько датасетов из репозитория `intellij-community`
- Получена решающая функция для регрессии для репозитория IntelliJ IDEA
- Разработан плагин для IntelliJ IDEA с встроенной решающей функцией



Спасибо за внимание