

Describe a Machine Learning task

Ich arbeite als Software-Entwickler bei einer großen österreichischen Bank. In meinem Team arbeite ich an einem Frontend Framework, welches innerhalb des Unternehmens von einigen Teams verwendet. Daher bekommen wir im Team von AnwenderInnen und EntwicklerInnen, welche mit der Software arbeiten, einige Bug Reports.

Problembeschreibung

Dieser Machine Learning Task soll Bug Reports anhand von diversen Kriterien automatisch klassifizieren. Oftmals sind Bug Reports schlecht formuliert und es fehlen essenzielle Informationen, die zur Problembehandlung notwendig sind. Das Überprüfen von Bug Reports kostet viel Zeit und ist immer wieder mit Rückfragen verbunden, welche zeitaufwändig ist. Der Machine Learning Task soll im ersten Schritt prüfen, ob die eingehenden Bug Reports vollständig sind und alle notwendigen Informationen enthält.

Jeder Bug Report braucht unbedingt einen Titel sowie eine kurze Beschreibung des Problems. Insgesamt gibt es zwei verschiedene Arten von Bug Reports. Zum einen können Bug Reports von AnwenderInnen einer Software und zum anderen von EntwicklerInnen, die ein firmeninternes Programmierframework verwenden, erstellt werden. Je nach Art des Bug Reports gibt es unterschiedliche Pflichtfelder. Jeder Bug Report muss auch eine Anleitung beinhalten, wie man den Bug reproduzieren kann, sowie ein Bild bzw. Gif des beschriebenen Verhaltens. Im Screenshot muss der komplette Screen der Applikation abgebildet sein, ansonsten wird der Screenshot nicht akzeptiert. Des Weiteren muss der Name und die Version der Software eingetragen werden.

Ein Bug Report von EntwicklerInnen muss zusätzlich noch einen Link zum Source Code des Projekts enthalten, sowie ein eigenständiges Mini-Repository, welches auf den problematischen Code reduziert wurde, damit dieses ohne weiteres ausgeführt werden kann. Des Weiteren muss bei einem/r DeveloperIn der Programmcode, wo der Fehler auftritt, angegeben werden. Dieser Vorgang dient dazu, dass der Bug von BearbeiterInnen des Bug Reports einfacher nachgestellt werden. Dadurch kann ausgeschlossen werden, dass der Bug durch ein Zusammenspiel von verschiedenen Komponenten aufgetreten ist. EntwicklerInnen müssen ebenfalls Informationen zu ihrer Entwicklungsumgebung angeben, wozu beispielsweise die Version der Laufzeitumgebung der Programmiersprache, sowie die Version des internen Frameworks zählt. Wenn nicht alle notwendigen Informationen enthalten sind, dann wird der Bug Report abgewiesen und der/die Ersteller/in wird darüber informiert, was noch ergänzt werden muss.

Arbeitsschritte

Wenn der Bug Report laut Task vollständig ist, dann klassifiziert der Machine Learning Task aufgrund des Screenshots, in welcher Applikation der Fehler aufgetreten ist. Zudem wird bestimmt, welches Team für den betroffenen Teil der Applikation zuständig ist. Der Bug Report wird dann an das zuständige Team weitergeleitet.

Die Daten, welche bereits zur Verfügung stehen, sind alte vollständig bearbeitete Bug Reports. Anhand dieser Bug Reports kann der Algorithmus trainiert werden, welche Informationen einen Bug Report vollständig machen. Es können ebenfalls Bug Reports, welche nicht vollständig waren, herangezogen werden. Dadurch kann dem Algorithmus gezeigt werden, welche Informationen ein Bug Report unbedingt enthalten muss. Ebenfalls können bereits vorhandene Bug Reports noch verbessert werden, falls Informationen fehlen. Anhand dessen existieren sehr viele Daten für den Machine Learning Task.

Der Algorithmus soll auch erkennen, ob die angegebene URL des Repository des/der EntwicklerIn auf eine Source Code Datei mit Zeilennummer zeigt. Wenn dies nicht der Fall ist, soll der Bug Report abgewiesen werden.

Der Algorithmus soll aus dem Screenshot der Applikation erkennen können, welche Applikation von diesem Bug Report betroffen ist. Des Weiteren soll erkannt werden, in welchem Teil der Applikation das Problem aufgetreten ist, damit aufgrund dessen das zuständige Team informiert werden kann. Der Algorithmus soll aus dem Header bzw. aus dem Footer im Screenshot die Applikation, sowie die Version der Applikation identifizieren. Anhand der sichtbaren Seiten Navigation im Screenshot kann erkannt werden, welche Seite konkret einen Fehler ausgibt oder nicht richtig funktioniert. Falls die Erkennung aus dem Screenshot nicht funktioniert, muss der/der Reporter/In einen neuen Screenshot hochladen, wo die notwendigen Informationen sichtbar sind. Damit dieses Verhalten umgesetzt werden kann, muss jede Anzeigemaske der Software durch ein automatisiertes Tool aufgezeichnet werden und mit Schlagworten versehen werden. Dies könnte durch einen anderen Machine Learning Task verarbeitet werden, der die verschiedenen Screenshots der Applikation klassifiziert.

Des Weiteren muss noch erfasst werden, welches Team für welche Applikation bzw. Teile einer Applikation zuständig ist, damit der Machine Learning Task den Bug Report dem richtigen Team zuweisen kann. Die Teams bekommen den vollständigen Report dann in ihre Inbox und sehen alle wichtigen Informationen sofort auf einen Blick.

Der Machine Learning Task könnten potenziell auch feststellen, ob so eine ähnliche Anfrage bereits gestellt wurde. Bug Reports müssen nicht zwangsläufig einen Bug beschreiben. Das beschriebene Verhalten kann auch den Vorgaben der Applikation entsprechen. Wenn der Algorithmus erkennt, dass das beschriebene Verhalten einem bereits bearbeiteten Bug Report gleicht, dann bekommt der/die BenutzerIn den ähnlich abgeschlossenen Bug Reports vorgelegt und kann prüfen, ob es dem beobachteten Verhalten entspricht. Wenn ja, dann muss in diesem Fall kein Bug Report erstellt werden und das Team wird entlastet.

Weitere Daten

In die Applikationen könnte in Zukunft ein Tool integriert werden, welches die BenutzerInnen dabei unterstützt einen Bug Report zu erstellen. Falls den BenutzerInnen ein Bug unterkommt, klicken diese einfach auf einen Button in der Applikation und ein Fenster öffnet sich, wo eine Beschreibung und ein Titel eingegeben werden kann. Alle anderen Informationen, beispielsweise der Screenshot werden automatisch angehängt, sodass sich der/die BenutzerIn nur auf die Beschreibung konzentrieren kann. Auch für EntwicklerInnen wäre so ein ähnliches Tool hilfreich. Beispielsweise könnte dieses Tool in die

Entwicklungsumgebung als Plugin eingebunden werden. Dadurch kann der/die EntwicklerIn auf die fehlerhafte Stelle im Code klicken und einen Bug Report daraus erstellen. Danach muss diese/r auch nur den Titel und die Beschreibung des Bugs eingeben. Alle anderen Informationen für EntwicklerInnen Bug Reports wie das Repository können somit automatisch angehängt werden.

Eventuell könnten auch noch Daten aufgezeichnet werden, wie der/die AnwenderIn mit der Applikation interagiert. Anhand dieser können die Schritte zum Reproduzieren des Bugs einfach aus dem Tool erstellt werden.

Machine Learning Model

Ich würde für diese Aufgabe den Decision Tree als Classifier verwenden. Für den Decision Tree berechne ich die Precision, Accuracy und den Cross Value Score. In der letzten Data Science Aufgabe habe ich mit dem Decision Tree die besten Ergebnisse erzielt.

Als Trainingsdaten nehme ich 66% der vorhandenen Daten. Mit diesem Datenset trainiere ich den Algorithmus. Mit den restlichen 33% teste ich mein Modell.

Mithilfe der Ergebnisse kann ich die Qualität meines Modells bewerten. Stichprobenartig kontrolliere ich noch die Klassifikationen des Algorithmus.

Fazit

Dieser Machine Learning Task wäre eine riesige Erleichterung in meinem Arbeitsalltag. Beim Bearbeiten von Bug Reports könnte viel Zeit gespart werden. Ich denke, dass dieser Machine Learning Task mit den verfügbaren Daten im Unternehmen umgesetzt werden könnte. Auch zukünftige Erweiterungen des Tasks bringen viele Erleichterungen für EntwicklerInnen.