Software-Sprint: Prototype Fund

Algoneer

Schlussbericht

Zuwendungsempfänger: Andreas Dewes & Katharine Jarmul GbR

Dr. Andreas Dewes

Berlin, den 30.09.2019

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS18S59** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Algorithmische Systeme werden in vielen Bereichen eingesetzt, um Entscheidungen und Abläufe durch Software zu automatisieren. Oft werden hierfür Verfahren des maschinellen Lernens (ML) bzw. der künstlichen Intelligenz (KI) eingesetzt. Im Gegensatz zu herkömmlichen Algorithmen werden diese Verfahren dabei nicht manuell programmiert, sondern anhand von Daten auf ein gegebenes Problem „trainiert“. Moderne ML-Verfahren sind hochkomplex und selbst für die Programmierer der Systeme nicht einfach nachzuvollziehen. Dies schafft eine Reihe von Problemen beim alltäglichen Einsatz dieser Systeme: Mangelnde Nachvollziehbarkeit untergräbt Vertrauen der Benutzer und kann zu ernsten Problemen führen, beispielsweise wenn algorithmische Systeme falsche Entscheidungen treffen die Menschen persönlich einschränken, sich in kritischen Situationen falsch entscheiden, Personen aufgrund befangener Trainingsdaten diskriminieren oder die Systeme von außen manipulierbar sind. Bisher gibt es nur wenige Ansätze um ML-Systeme systematisch auf Robustheit, Sicherheit, Transparenz und Nachvollziehbarkeit zu testen. Da solche Systeme in immer mehr Bereichen eingesetzt werden stellt dies ein ernstes Problem für die Gesellschaft dar. Mit Algoneer wollen wir eine Lösung schaffen, die Entwicklern aber auch Betroffenen und Auditoren von ML-Systemen ermöglicht, diese besser zu verstehen und gemeinwohlorientiert zu gestalten. Unser Ansatz ist, eine Open-Source Software zu schaffen, die einfach in bestehende ML-Systeme integriert werden kann und Entwicklern hilft, diese kontinuierlich zu analysieren und zu überwachen. Die Software soll ermöglichen, aktuelle Forschungsergebnisse und Methoden aus der akademischen Forschung für die Untersuchung von ML-Verfahren einfach in der Praxis zu nutzen. Gleichzeitig sollen die Ergebnisse der

Testverfahren in einfacher und ansprechender Weise aufbereitet werden, so dass sie auch von technisch wenig erfahrenen Nutzern interpretiert werden können. Die Lösung soll sprach- und systemunabhängig funktionieren und über eine Web-Anwendung gesteuert werden können. Wir haben uns zu Beginn des Projekts folgende Meilensteine gesetzt:

- **Umsetzung des grundlegenden Backends mit Python und Docker (Workflow-System und API)**
- **Umsetzung des grundlegenden Frontends mit React.js**
- **Umsetzung eines ersten Beispielplugins (z.B. LIME)**
- **Regelmäßige Diskussion mit Stakeholdern (Data Scientists + Stiftungen / Organisationen)**
- **Whitepaper zu Methodik und Umsetzung**
- **Automatisiertes Deployment für das Projekt (Docker-Installer / Ansible-Setup)**
- **Dokumentation der Systemkomponenten**
- **Webseite für das Projekt**
- **Vorstellung auf relevanten Konferenzen**

Während des Projekts haben wir einzelne Meilensteine basierend auf gewonnenen Erkenntnissen angepasst, so haben wir das Backend z.B. nicht als Docker-basiertes System sondern als reine Python-Anwendung umgesetzt, wobei wir eine API sowie eine Python-Bibliothek zur einfachen Anbindung an unterschiedliche ML-Systeme implementiert haben.

Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“

Wir haben allgemein drei Zielgruppen für unsere Lösung definiert:

- **Entwickler von ML-Systemen**
- **Menschen, die mit ML-Systemen arbeiten oder von diesen betroffen sind**
- **Auditoren die ML-Systeme begutachten und bewerten sollen**

Im Rahmen des Prototype Fund (PF) Projektes haben wir die Gruppe der Entwickler priorisiert, da wir denken, dass diese Zielgruppe am stärksten von unserer Lösung profitieren kann. Zusätzlich haben wir uns bei den Testmethoden für ML-Verfahren auf Methoden für die Nachvollziehbarkeit dieser Verfahren beschränkt, um mit dem limitierten zeitlichen Umfang des Projektes ein relevantes Ergebnis erzielen zu können. Unsere Lösung bezieht sich thematisch daher sehr genau auf das Themenfeld des Software Sprints „Maschinen Lernen Lassen“ und wir sehen die erzielten Ergebnisse als sehr guten Beitrag, da die erstellte Software aktuell eine einfache und leicht einzusetzende Möglichkeit bietet, ML-Verfahren systematisch und kontinuierlich zu testen und zu überwachen. Unsere Software macht damit aktuelle Forschungsergebnisse in diesem Bereich einer breiteren Nutzerschicht verfügbar. Insbesondere im Rahmen der KI-Initiative der Bundesregierung und der europäischen Union

sehen wir Algoneer als eine wichtige Komponente zur Umsetzung von Nachvollziehbarkeit und Transparenz bei dem Einsatz algorithmischer Systeme.

Ausführliche Darstellung der Ergebnisse

Wir haben die uns gesteckten Ziele vollständig erreicht und zum Projektende im August einen funktionsfähigen Prototyp präsentiert, der alle wesentlichen geplanten Funktionen enthält und voll lauffähig ist. Im Rahmen der geplanten Meilensteine haben wir im Detail die folgenden Ergebnisse erzielt.

Umsetzung des grundlegenden Backends mit Python und Docker

Wir haben zwei grundlegende Backend-Komponenten implementiert: Eine Python-Bibliothek (algoneer), welche einfach in bestehende Python-basierte ML-Systeme integriert werden kann und in der Lage ist, mit unterschiedlichen Datenformaten und ML-Bibliotheken zu arbeiten. Benutzer können über die Bibliothek automatisiert verschiedene Tests auf ML-Verfahren anwenden (u.a. SHAP, PDP, ALE) und die Ergebnisse systematisch speichern und zu Dokumentationszwecken mit spezifischen Modellen, Datensätzen und Algorithmen assoziieren. Die Testergebnisse können direkt über die Bibliothek begutachtet werden oder über eine REST-Schnittstelle an die Algonaut-API gesendet werden, was eine Analyse über die Web-Anwendung ermöglicht. Wir haben neben drei implementierten Tests beispielhaft Schnittstellen für populäre Python-Bibliotheken zur Datenverwaltung (pandas) sowie für das maschinelle Lernen (scikit-learn) erstellt, die erläutern wie Algoneer an unterschiedliche ML-Technologien angepasst werden kann. Alle Schnittstellen wurden dabei so generisch wie möglich definiert, um eine hohe Anpassbarkeit zu gewährleisten, wir hoffen damit eine Anbindung an weitere ML-Bibliotheken einfach zu machen. Die Python-Bibliothek wurde mit einer modernen Python-Version erstellt (3.7) und setzt konsequent auf eine hohe Code-Qualität. Zur Sicherung der Qualität wurden für alle wesentliche Systemkomponenten ausführliche Unit-Tests definiert, zudem wurde die komplette Code-Basis mit Typinformationen annotiert.

Zusätzlich zur Python-Bibliothek haben wir eine Python Web-API erstellt, Algonaut, die eine REST-Schnittstelle zur Speicherung der Testergebnisse der Algoneer-Bibliothek bietet. Die API kann unabhängig von Algoneer betrieben werden und bietet eine Möglichkeit, Testergebnisse von Algoneer aber auch aus anderen Quellen systematisch zu erfassen und zu speichern. Die REST-API wurde genau wie die Python-Bibliothek mit modernen Methoden der Software-Entwicklung erstellt, verfügt über umfangreiche Unit-Tests und ist zudem über eine OpenAPI-Spezifikation vollständig dokumentiert. Die Dokumentation per OpenAPI-Standard ermöglicht zudem, in einer Vielzahl an Programmiersprachen automatisiert Schnittstellen zu der API zu definieren und so einfach Daten an diese zu senden.

Umsetzung eines ersten Beispielplugins (z.B. LIME)

Wir haben im Rahmen der Evaluation von Testmethoden für ML-Verfahren alle gängigen Methoden und Bibliotheken evaluiert, u.a. SHAP[1], LIME[2] sowie eine Reihe weiterer Verfahren[3][4]. Wir haben exemplarisch drei dieser Verfahren in unserer Algoneer-Bibliothek implementiert. Die Einbettung in Algoneer macht die Verfahren dabei einfach nutzbar da eine Konfiguration der einzelnen Tests durch den Nutzer entfällt. Vielmehr muss der Nutzer lediglich einmalig die Datenstruktur der Testdaten definieren und kann dann alle aufgeführten Tests automatisiert ausführen. Dies spart gerade bei der Entwicklung von ML-Verfahren wertvolle Entwicklerzeit und macht Testergebnisse robuster und systematisch nachvollziehbar.

Regelmäßige Diskussion mit Stakeholdern

Im Rahmen des Projektes haben wir mit 15 Organisationen detaillierte Interviews geführt, u.a. mit privaten Unternehmen, zivilgesellschaftlichen Organisationen, Stiftungen aber auch Privatpersonen und Gewerkschafen. Basierend auf diesen Interviews haben wir die Feature-Entwicklung priorisiert und angepasst und haben sehr viel wertvolles Feedback zur Gestaltung einer relevanten Lösung von unseren Zielgruppen erhalten.

Whitepaper zu Methodik und Umsetzung

Wir haben die Ergebnisse des Projekts im Rahmen eines ausführlichen Blog-Posts zusammengefasst und haben diesen auf unserer Webseite publiziert.

Automatisiertes Deployment für das Projekt

Wir haben basierend auf Nutzerfeedback ein komplexes Deployment unserer Lösung komplett vermieden. Die erstellte Software kann vielmehr in wenigen Schritten als Python-Bibliothek installiert und genutzt werden, ein komplexer Deployment-Prozess ist somit nicht erforderlich. Wir haben die Installation der einzelnen Systemkomponenten (Algoneer, Algonaut und Frontend) ausführlich dokumentiert.

Dokumentation der Systemkomponenten

Wir haben für die Algoneer Python-Bibliothek eine ausführliche Dokumentation mithilfe des „Sphinx“ Dokumentationssystems erstellt. Für die Algonaut API haben wir ebenfalls eine vollständige Dokumentation aller Endpunkte mithilfe der OpenAPI-Spezifikation erstellt. Das Frontend wurde im Code selbst dokumentiert, die Nutzung ist weitgehend selbsterklärend und erfordert keine ausführliche Dokumentation.

[1] https://arxiv.org/abs/1705.07874
[2] https://arxiv.org/abs/1602.04938
[3] https://christophm.github.io/interpretable-ml-book/ale.html
[4] https://christophm.github.io/interpretable-ml-book/pdp.html

Webseite für das Projekt

Wir haben eine statische Website mithilfe eine Page-Building Tools erstellt und auf https://algoneer.org veröffentlicht. Die Inhalte der Webseite wurden mehrfach angepasst und erweitert basierend auf Feedback, das wir während der Projektphase erhalten haben.

Vorstellung auf relevanten Konferenzen

Wir haben Algoneer u.a. auf lokalen Veranstaltungen in Berlin (z.B. bei Gewerkschaften) als auch in virtuellen Konferenzen (z.B. im Rahmen von Bitkom-Events) vorgestellt. Wir haben im Rahmen einer Bitkom-Publikation (Erscheinungsdatum Oktober 2019) den Ansatz von Algoneer vorgestellt und exemplarische Tests erläutert, die für die Nachvollziehbarmachung von ML-Verfahren genutzt werden können.

Wir haben im Rahmen der Arbeit an dem Projekt unser Verständnis der Notwendigkeit einer Lösung für das Testen von ML-Verfahren stark konkretisiert und erste klare Marktbedürfnisse identifiziert. Wir planen, den im Rahmen des PF erstellten Prototypen kontinuierlich weiterzuentwickeln und gegebenenfalls im Rahmen einer professionellen Lösung anzubieten. Die Förderung hat uns dabei geholfen, die nötige Marktrecherche und Interviews sowie die Entwicklung des Prototyps zu finanzieren. Die Unterstützung durch das Prototype Fund Team war insbesondere zu Anfang des Projektes hilfreich. Ohne die Förderung des PF und BMBF hätten wir das Projekt nicht realisieren können.

Umsetzung einer Web-Anwendung

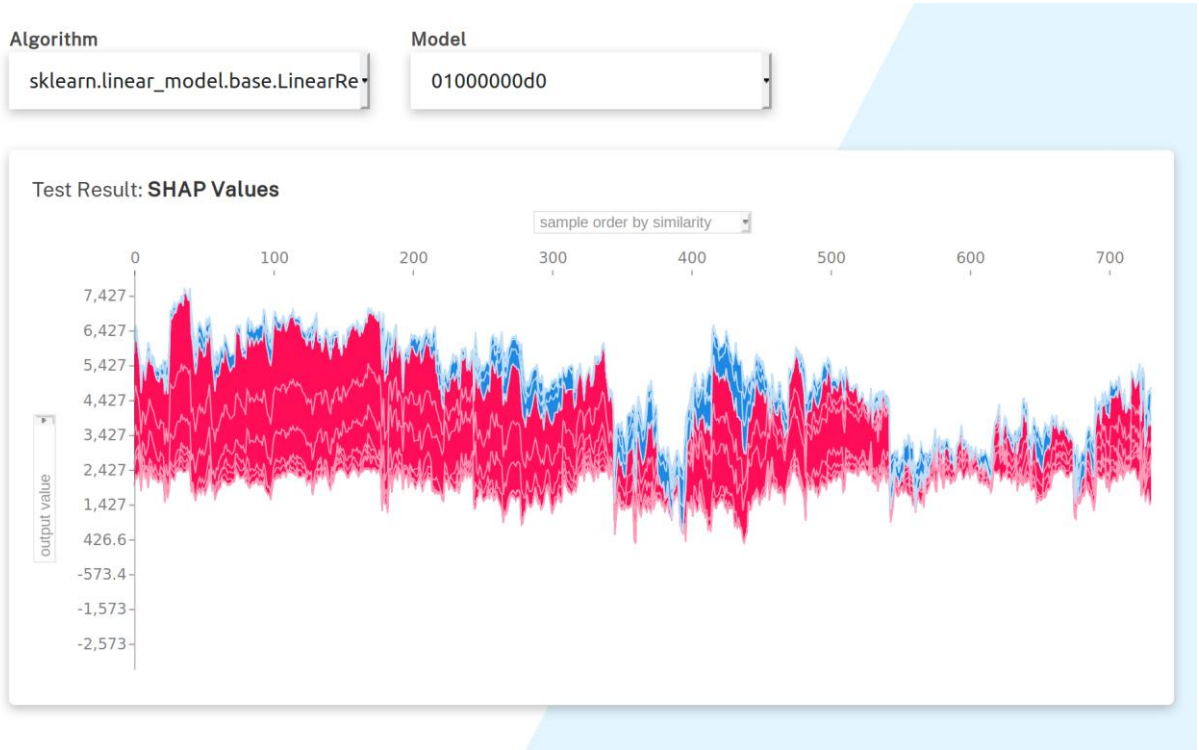


Abb. 1: Algoneer Web-Anwendung. Beispiel-Ergebnis eines SHAP-Tests für ein lineares ML-Modell.

Um die mit Algoneer erstellten Testergebnisse möglichst einfach nutzbar und teilbar zu machen, haben wir eine Web-Anwendung erstellt. Diese Anwendung nutzt die REST-API von Algonaut, um Testergebnisse von einem Server abzurufen und darzustellen. Wir haben im Rahmen des Projektes exemplarisch Testergebnisse eines einzelnen Verfahrens (SHAP) in der Web-Anwendung implementiert, was eine einfache Auswertung dieser Ergebnisse für den Nutzer ermöglicht. Die Web-Anwendung ist Multi-User fähig und kann mit verschiedenen Authentifizierungs-Lösungen betrieben werden, wir haben exemplarisch Logins via Github, Gitlab, Google sowie E-Mail und Passwort angebunden. Nutzer können in der Software Organisationen anlegen und andere Nutzer zu ihren Organisationen einladen, Testergebnisse können so leicht in Teams geteilt und nachvollzogen werden. Abb. 1 zeigt einen Screenshot der Anwendung, welcher das Ergebnis des SHAP-Tests für ein Beispiel ML-Verfahren zeigt.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Wir haben mit Algoneer gezeigt, dass automatisiertes Testen von ML-Verfahren möglich und sinnvoll ist, und den Aufwand hierfür für Anwender drastisch reduziert. Wir erlauben damit Entwicklern von ML-Verfahren, ihre Systeme kontinuierlich und systematisch zu testen und die Testergebnisse mit allen relevanten Stakeholdern zu teilen. Wir planen, die Lösung sowohl durch die Anbindung weiterer Testmethoden als auch durch die Unterstützung zusätzlicher ML-Bibliotheken zu erweitern. Zusätzlich planen wir den Betrieb einer SaaS-Lösung und möglicherweise die Schaffung einer professionellen Lösung basierend auf unserem Open-Source Tool. Wir planen aktuell ca. 2 Personentage pro Woche für die Weiterentwicklung von Algoneer zu nutzen und weiterhin kontinuierlich Feedback für die Lösung einzusammeln. Wir sind zudem in Gesprächen mit verschiedenen Organisationen und Unternehmen, die Interesse gezeigt haben, Algoneer in der Praxis zu nutzen, um ML-Verfahren zu testen und zu überwachen. Die Tatsache, dass unsere Software als Open-Source Lösung frei verfügbar ist und flexibel genutzt werden kann macht die konkrete Nutzung sehr einfach. Dies ermöglicht z.B. Unternehmen, Organisationen und Privatpersonen, die Software ohne hohe Risiken einzusetzen und einfach an ihre Bedürfnisse anzupassen.

Die Arbeit an dem Projekt hat uns persönlich in unserer fachlichen Entwicklung stark vorangebracht. Wir haben heute ein sehr viel besseres Verständnis über die Gestaltung und die Überwachung von ML-Verfahren und haben zudem unsere Kenntnisse in moderner Software-Entwicklung und nutzergetriebener Produktgestaltung ausbauen können. Wir sind auf gutem Weg, Experten im Bereich sicherer, nachvollziehbarer KI zu werden und sehen basierend hierauf viele Möglichkeiten, unsere im Rahmen des Projekts gewonnenen Kenntnisse in der Praxis weiter einzusetzen.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Wir haben ursprünglich geplant, das Analyse-Backend mit „Docker“ zu realisieren, dies haben wir zu Gunsten einer einfacheren Lösung verworfen und haben stattdessen eine reine Python-Bibliothek implementiert. Dies hat sich rückblickend als der richtige Ansatz herausgestellt und hat uns zudem erlaubt, schnell erste Ergebnisse zu erzielen und diese für Feedback von Nutzern einzusetzen. Die Gestaltung des Frontends haben wir im Laufe des Projekts ebenfalls stark vereinfacht, was uns erlaubt hat schneller eine funktionsfähige Version zu veröffentlichen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Wir haben verschiedene Materialien erstellt, die potenziellen Nutzern erlauben mehr über Algoneer zu erfahren:

- Unsere Website https://algoneer.org (Abb. 2) enthält wesentliche Informationen über unser Projekt und die Zielgruppen sowie Einsatzmöglichkeiten, sie dient damit als erste Anlaufstelle für potenzielle Nutzer.
- Unsere Github-Seite https://github.com/algoneer enthält alle Software-Komponenten, die während des Sprints erstellt wurden und erlaubt Nutzern, einen schnellen Überblick über die Installation und Nutzung dieser Komponenten zu erhalten. Zusätzlich enthalten einzelne Repositories Beispiele (z.B. https://github.com/algoneer/algoneer/tree/master/examples) zu Testmethoden, die von Algoneer unterstützt werden. Aktuell befinden sich auf Github die Projekte „algoneer“ (Python-Bibliothek), „algonaut“ (Python-basierte REST API), „algoneer-app“ (Algoneer Web-Anwendung / Frontend), „datasets“ (Beispieldatensätze die zum Testen von Algoneer eingesetzt werden), „website“ (Algoneer-Webseite) und „algonaut-plugins“ (Plugins für die Algonaut REST-API).
- Unsere Gitlab-Seite https://gitlab.com/algoneer enthält die gleichen Komponenten wie unsere Github-Seite, zusätzlich werden hier jedoch automatisierte Tests bei jeder Code-Änderung ausgeführt. Diese Seite erlaubt die Zusammenarbeit mit Entwicklern, die uns bei der Weiterentwicklung von Algoneer unterstützen wollen.
- Unsere Dokumentation https://docs.algoneer.org enthält detaillierte Angaben zu den einzelnen Systemkomponenten und zeigt exemplarisch deren Nutzung.
- Der Prototype Fund hat freundlicherweise zwei Videos produzieren lassen, die unser Algoneer Projekt vorstellen bzw. allgemein über die 5. Wettbewerbsrunde berichten: https://www.youtube.com/watch?v=xLJtrilzwUE sowie https://www.youtube.com/watch?v=MpFtAmz7UTQ. Das erste Video ist ebenfalls auf unserer Webseite verfügbar.
- Im Rahmen einer Bitkom-Publikation zum Thema „nachvollziehbare KI-Methoden“ haben wir einen ausführlichen, mehrseitigen Abschnitt über Algoneer sowie

Testmethoden für nachvollziehbare ML-Verfahren im Allgemeinen beigetragen. Die Publikation wird voraussichtlich im Oktober 2019 erscheinen.

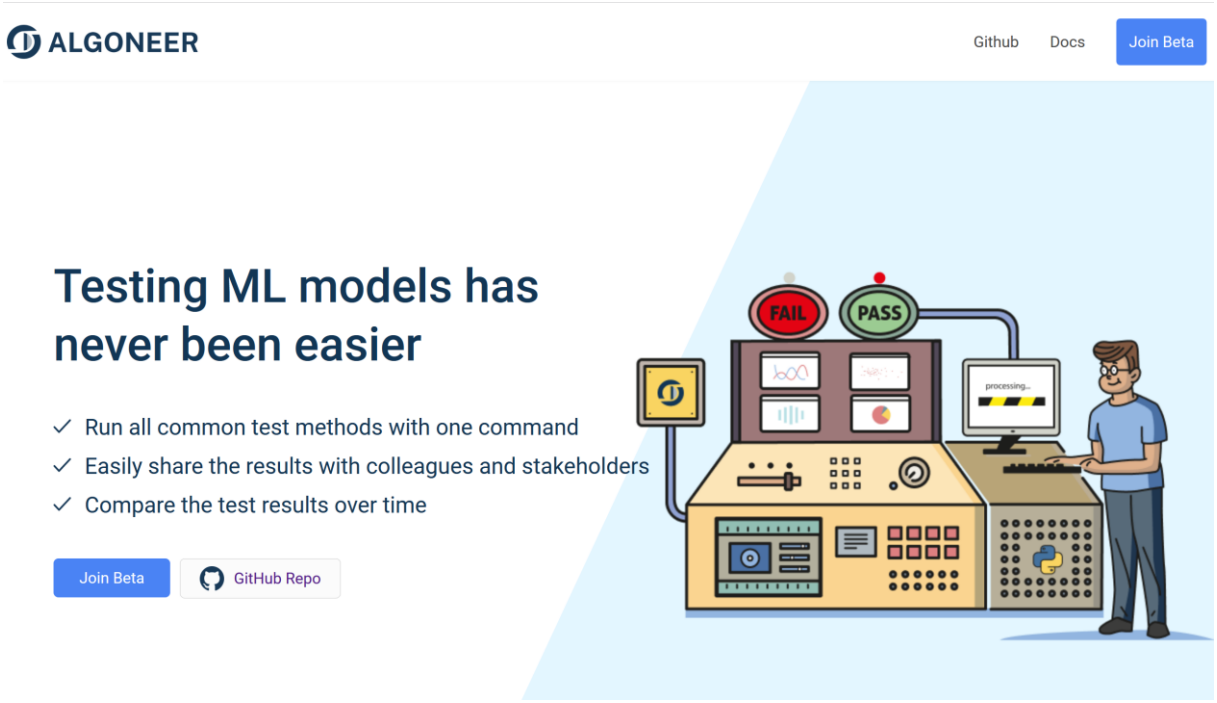


Abb. 2: Algoneer-Webseite (Stand September 2019)

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Es hat sich herausgestellt, dass Projektmanagement, Interviews mit potenziellen Nutzern sowie Forschungs- und Recherchearbeiten einen höheren Aufwand verursacht haben als ursprünglich geplant. Zusätzlich mussten durch den Ausfall von Frau Jarmul, die sich leider nicht an dem Projekt aktiv beteiligen konnte, die Arbeiten alleinig von Andreas Dewes durchgeführt werden, was eine Umplanung erforderte, die jedoch machbar war.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Seit Einreichung der Bewerbung im Herbst 2018 gab es eine Reihe von Publikationen und Veröffentlichungen im Bereich nachvollziehbarer ML-Verfahren, u.a. haben IBM und andere große Stakeholder Bibliotheken veröffentlicht, um ML-Systeme beispielsweise auf Fairness zu untersuchen. Publikationen wie das „Interpretable ML Book“ von Christoph Molnar[5] haben zudem die Recherche und Implementierung geeigneter Testverfahren für uns stark vereinfacht, was unser Projekt sogar beschleunigt hat. Insgesamt haben wir gesehen, dass das Interesse an Verfahren für das Testen von ML-Systemen im letzten Jahr sprunghaft angestiegen ist, was sicherlich mit der starken Thematisierung von künstlicher Intelligenz im Allgemeinen sowie den zahlreichen Bundesinitiativen in diesem Bereich zusammenhängt.

[5] https://christophm.github.io/interpretable-ml-book/pdp.html

Insgesamt sehen wir diese Entwicklung als sehr positiv und denken, dass wir mit Algoneer genau zum richtigen Zeitpunkt ein Tool schaffen, das ein zunehmend relevantes Problem auf einfache Art löst. Bisher hat sich noch keine größere Entwickler- und Nutzer-Community um Algoneer gebildet, wir planen jedoch für Oktober den offiziellen Launch einer gehosteten Version der Software und damit einhergehend auch größere Aufmerksamkeit von seitens der Entwickler-Community sowie möglicher Nutzer. Die große Popularität des Themas KI/ML sollte uns hierbei eher helfen.

Wir sind sehr dankbar für die erhaltene Förderung und die Betreuung durch den Prototype Fund und hätten Algoneer in der aktuellen Form ohne diese Förderung niemals realisieren können.

Berlin, den 30.09.2019

Dr. Andreas Dewes