

Optimind

Optimind

Programm / Ausschreibung	IWI 24/26, IWI 24/26, Basisprogramm Ausschreibung 2025	Status	laufend
Projektstart	21.10.2024	Projektende	20.10.2025
Zeitraum	2024 - 2025	Projektlaufzeit	13 Monate
Keywords			

Projektbeschreibung

Ziele und Ergebnisse:

Ziel unseres Projekts ist es, ein KI-Modell zu entwickeln, das menschliche Psychologie sowie Entscheidungsprozesse versteht und dadurch Verkaufstexte schreiben kann, die außergewöhnlich gut funktionieren. Unsere Software soll vollautomatisiert Verkaufsseiten im Internet optimieren. Das bedeutet, sie soll Seiten analysieren und vorhersagen, wo die größten Potenziale für mehr Conversion-Rate sind, für diese Elemente neue Variationen generieren, direkt testen, was besser funktioniert, mit den Ergebnissen lernen und gleich wieder neue Variationen generieren. Dabei soll das Modell branchen- und produktunabhängig funktionieren und sich an verschiedene Zielgruppen anpassen können.

Um unsere Ziele zu erreichen, werden wir ein Basis-KI-Modell mit einer A/B-Testing Architektur verbinden, ein tiefes Fine Tuning vornehmen und das Modell im Live-Betrieb laufend trainieren.

Die Software soll nach Fertigstellung folgende Performance-Kriterien erfüllen:

- 90% der Elemente auf einer Landing Page erkennen.
- Die vorhergesagte Leistungssteigerung des Modells soll auf $\pm 10\%$ genau mit der realen Steigerung übereinstimmen.
- Das Modell soll Variationen generieren, die mit einem Signifikanzniveau von 5% besser funktionieren als die Control-Variante.
- 80% der gesamten Conversions soll das Modell den Variationen zuordnen können.

Um schnell auf den Markt zu kommen und frühzeitig Feedback für Weiterentwicklungen unserer KI zu sammeln, wollen wir mit einem Minimum Viable Product (MVP) starten, das gerade genug Features hat, um Kernfunktionalitäten zu liefern und den Nutzern einen Mehrwert zu bieten, während es gleichzeitig ein Minimum an Entwicklungszeit und -ressourcen erfordert. Nach der erfolgreichen Einführung werden wir die nächste Version mit erweiterten Funktionalitäten entwickeln.

In der ersten Phase geht es darum, ein Modell zu entwickeln, das hinreichend gute Ergebnisse für Textbausteine liefert, um sie in A/B-Tests verwenden zu können – das Modell soll Verkaufstexte verstehen lernen. In der zweiten Phase soll durch Performance Reinforced Learning eine Softwarelösung entwickelt werden, die automatisch für Landingpages neue Varianten von Seitenelementen generiert, die Varianten gegeneinander testet, damit laufend die Performance der Seite verbessert und das zugrundeliegende Modell auf Basis der Ergebnisse optimiert. Sobald die Architektur funktioniert und gute Ergebnisse mit dem Testen von Headlines erzielt werden, wird das Modell in Phase drei auf die restlichen Elemente einer Landingpage

ausgeweitet. Wenn alle Seitenelemente unterstützt werden, soll die Softwarelösung eigenständig entscheiden, welches Seitenelement als nächstes am sinnvollsten ist und wahrscheinlich die besten Ergebnisse liefert. Anschließend sollen nach und nach neue Arten von Seiten hinzugefügt und unterstützt werden. Das Optimieren des Modells findet pro Kunde statt, pro Seitenelement, pro Seitentyp, aber auch über das gesamte Modell und alle Kunden hinweg. Sobald alle Seitenelemente einer Landingpage und verschiedene Arten von Seiten unterstützt werden, das Modell vollautomatisch Varianten generiert, testet und von den Ergebnissen lernt, werden in Phase vier Zugänge für das Modell verkauft.

Am Ende der Projektlaufzeit soll das Modell Landingpages und deren Elemente erkennen und ansteuern. Das Modell soll außerdem vorhersagen können, welche Elemente das größte Potenzial haben. Es soll Variationen dieser Elemente generieren können, die signifikant besser funktionieren als die Control-Variante. Des Weiteren soll das Modell Split-Tests durchführen und auswerten können inklusive Conversion-Tracking und verschiedene Arten von Verkaufsseiten optimieren. Grundsätzlicher Lösungsansatz:

Der innovative Ansatz dieses Projekts ist die Kombination von Performance-Daten mit einem Open Source Large Language Modell. Wir werden das Modell mit einem bestehenden Datensatz aus erfolgreichen Verkaufsseiten finetunen, sodass wir hinreichend gute Ergebnisse erzielen, um mit den A/B Tests zu starten. Die A/B Tests werden über ein JavaScript Snippet auf der zu optimierenden Verkaufsseite und der Conversion-Seite durchgeführt. Das Snippet übermittelt die Verkaufsseite an das KI-Modell, das KI-Modell liefert die zu testenden Varianten, der Code teilt den Traffic auf und setzt die Varianten in die Seite ein. Während dem Test wird gemessen, welche Nutzer konvertieren und welche Variante sie gesehen haben. Am Ende wird die beste Variante als Control übernommen und der nächste Test mit dem Wissen der vorherigen Tests gestartet. Die Architektur wird so aufgesetzt, dass wir jederzeit das Basismodell austauschen können, da im Bereich der KI laufend große Verbesserungen veröffentlicht werden.

Risiken:

Die Wirkung unserer Lösung hängt stark von der genauen Erkennung der Landingpage-Elemente ab. Daher wird ein Classifier-Modell mit Human Feedback Reinforced Learning (HFRL) entwickelt, um Landingpages präzise von anderen Zielseiten zu unterscheiden. Dazu muss es CSS-Selektoren präzise identifizieren, um die Elemente korrekt zu klassifizieren. Eine falsche Klassifizierung führt zu schlechten Optimierungen. Lösung 1: Eingabe des Element Textes in der Anwendung; Lösung 2: Direkt Eingabe des CSS Selektors in der Anwendung

Sobald das Modell Landingpages und deren Elemente erkennen kann, werden Werbebibliotheken durchsucht und samt Kennzahlen gespeichert. Diese Daten dienen dem Finetuning des LLMs, wobei der Input aus Landingpages als HTML-Dateien besteht. Aber KI-Modelle unterscheiden sich grundlegend von herkömmlich programmierten Programmen, da man nicht einfach den Ablauf anpassen kann, um das Ergebnis vorhersehbar zu verändern. Ein konkretes Risiko besteht darin, dass das Modell keine hochwertigen Headlines auf Basis von HTML-Seiten generiert, weil es möglicherweise nicht genügend Kontext oder relevante Informationen aus der HTML-Struktur und den Inhalten extrahieren kann. Wenn das Modell keine guten Headlines liefert, wird die Gesamtleistung der KI und ihre Fähigkeit, Verkaufsseiten zu optimieren, stark beeinträchtigt. Dieses Risiko könnte das Projekt zum Scheitern bringen, da die Generierung überzeugender Headlines ein zentraler Bestandteil des Erfolgs unserer KI-Software ist. Zur Qualitätsverbesserung werden verschiedene Lösungswege schrittweise eingesetzt. So kann jede Methode einzeln getestet und bewertet werden, bevor sie in den gesamten Workflow integriert wird. Die schrittweise Implementierung verbessert die Kostenkontrolle und sorgt für eine systematische Optimierung des Modells. Lösung 1: Prompt Engineering; Lösung 2: Anderes Modell (Mistral Nemo, Llama 3.1 8b); Lösung 3: Modell finetunen mit Copywriting Lektüre; Lösung 4: Input erweitern; Lösung 5: Modell finetunen mit guten Landingpages; Lösung 6: Modell finetunen mit den Ergebnissen aus manuellen A/B Tests; Lösung 7: Größeres Modell verwenden (Mistral 2 Large, Llama 70b, Llama 405b); Lösung 8: Input um händische Eingaben erweitern (Produktbeschreibung, USP, About Us, etc); Lösung 9:

Andere Sprachen

Für einen optimalen Lerneffekt des Modells brauchen wir ein präzises Tracking der Nutzer in den Tests. Das wird jedoch immer schwieriger. Wenn das Tracking von A/B Tests mit JavaScript nicht genau genug funktioniert, planen wir schrittweise Lösungsansätze anzuwenden, die die Genauigkeit verbessern und die Zuverlässigkeit der gesammelten Daten erhöhen. Falsche Daten führen zu einer mangelhaften Analyse der A/B-Testergebnisse, wodurch ineffektive Varianten als erfolgreich oder effektive Varianten als ineffektiv eingestuft werden können, was die gesamte A/B-Test-Methodologie untergräbt. Dieses Risiko könnte das Projekt zum Scheitern bringen, da ungenaue Messungen der A/B-Test-Ergebnisse zu falschen Schlussfolgerungen führen und die Optimierung der Verkaufsseiten durch die KI-Software ineffektiv machen würden. Jede Lösungsmethode hat unterschiedliche Auswirkungen auf die Genauigkeit der Daten und die Komplexität der Implementierung. Eine schrittweise Herangehensweise ermöglicht es, die Effektivität jeder Lösung zu bewerten, bevor weitere Maßnahmen ergriffen werden, und stellt sicher, dass das System nicht übermäßig komplex oder anfällig für Fehler wird. Lösung 1: First Party Cookies; Lösung 2: Finger Printing; Lösung 3: Google Analytics Integration; Lösung 4: Server to Server Tracking; Lösung 5: First Party Data; Lösung 6: Offline Conversion Tracking

Für das Conversion Tracking und um Seiten zu optimieren, muss der Kunde ein Javascript Snippet auf der zu optimierenden Seite platzieren. Anschließend generiert das Modell automatisch neue Varianten für die verschiedenen Seitenelemente. Diese Varianten werden in A/B-Tests gegeneinander und gegen die originale Variante getestet. Dabei werden Nutzer beim ersten Aufrufen der Landingpage per Zufall in Gruppen unterteilt und das Seitenelement je nach Gruppe per Javascript ausgetauscht. Falls der Nutzer erneut auf die Landingpage kommt, soll er wieder die gleiche Variante sehen. Anschließend wird getrackt, welche Nutzer das Ziel der Landingpage (Conversion) erreichen. Sobald genügend Daten für ein signifikantes Ergebnis zur Verfügung stehen, wird der Test beendet und das Ergebnis verwendet, um das Modell weiter zu verbessern. Wenn die Page Load Time beim Austauschen der Varianten in den A/B Tests zu lange dauert, kann das die Absprungrate erhöhen, wodurch sich die Testergebnisse verzerren. Die Optimierung der Ladezeit wird schrittweise umgesetzt, um sicherzustellen, dass die Nutzererfahrung nicht beeinträchtigt wird und die Testergebnisse präzise bleiben. Lösung 1: CDN aufsetzen; Lösung 2: Asynchrone Implementierung des JavaScript-Snippets

Auf Basis des Modells wird eine Softwarelösung entwickelt. Diese generiert automatisch für bestehende Landingpages neue Varianten von Seitenelementen, testet die Varianten gegeneinander, verbessert damit laufend die Performance der Seite und optimiert das zugrundeliegende Modell auf Basis der Ergebnisse (Performance Reinforced Learning). Allerdings beeinträchtigt regelmäßiges und umfangreiches Retraining durch hohe Rechenkosten und längere Rechenzeiten die Effizienz und Wirtschaftlichkeit des Projekts (wirtschaftliches Risiko ist ein technologisches Risiko!). Die Lösungen zur Reduktion der Rechenkosten und zur Verbesserung der Effizienz des Modells werden schrittweise umgesetzt. Eine schrittweise Implementierung ermöglicht es, die Auswirkungen jeder Maßnahme zu evaluieren und Anpassungen vorzunehmen, bevor weitere Änderungen eingeführt werden. Lösung 1: Trainingsintervall verlängern; Lösung 2: Ergebnisse mehr über Kontext verbessern und weniger über Finetuning; Lösung 3: Immer nur die neuesten Erkenntnisse trainieren (Chunked Finetuning) Reinforced Learning soll das Modell durch Rückmeldungen kontinuierlich verbessern. Wenn das Reinforced Learning die Ergebnisse nicht verbessert, wird die Optimierung des Modells stark beeinträchtigt und die Vorhersagefähigkeiten stagnieren oder verschlechtern sich. Wenn das RL-Modell nicht in der Lage ist, die richtigen Schlüsse aus den Daten zu ziehen und sich durch neue Variationen zu verbessern, wird das System seine Kernfunktion – die vollautomatisierte Optimierung von Verkaufsseiten – nicht erfüllen können. Lösung 1: Hyperparameter tunen; Lösung 2: Größere Datenmenge sammeln; Lösung 3: Trainings Features erweitern (zb. Branche, Conversion Art, etc); Lösung 3: Verschiedene Methoden (Kontext vs. Finetuning) probieren; Lösung 4: Datenqualität erhöhen; Lösung 5: Die Modellarchitektur ändern (Ein Varianten Modell und ein Prediction Modell vs. Fokus auf ein großes Variantenmodell); Lösung 6: Mehr Datenpunkte sammeln

Sobald die Architektur stabil ist und vielversprechende Ergebnisse beim Testen von Headlines liefert, wird das Modell auf die restlichen Elemente einer Landingpage ausgeweitet. Sobald alle Seitenelemente unterstützt werden, entscheidet die Software eigenständig, welches Element als nächstes für die A/B-Tests optimiert werden soll. Ungenaue Vorhersagen des Modells gefährden jedoch die Optimierung der Landing Pages. Da das Kernversprechen des Projekts auf der Automatisierung und der intelligenten Vorhersage zur Optimierung von Verkaufsseiten basiert, würde die Nichteinhaltung dieses Versprechens das Geschäftsmodell beeinträchtigen. Lösung 1: Hyperparameter tunen; Lösung 2: Daten anders aufbereiten; Lösung 3: Andere Inputs verwenden; Lösung 4: Trainings Features erweitern; Lösung 5: Größere Datenmengen sammeln

Die hohen laufenden Kosten großer KI-Modelle sind eine wirtschaftliche Herausforderung. Verzögert sich die Entwicklung und können wir keine ersten Kunden gewinnen, wird die Finanzierung der Hardware schwierig. Daher haben wir die Entwicklung so geplant, dass die Kosten bis zur Veröffentlichung des MVPs im Rahmen bleiben. Der Server kann flexibel ein- und ausgeschaltet werden, und wir optimieren die Serverkosten durch einen spezifischen KI-Use-Case.

Projektpartner

- Optimind GmbH