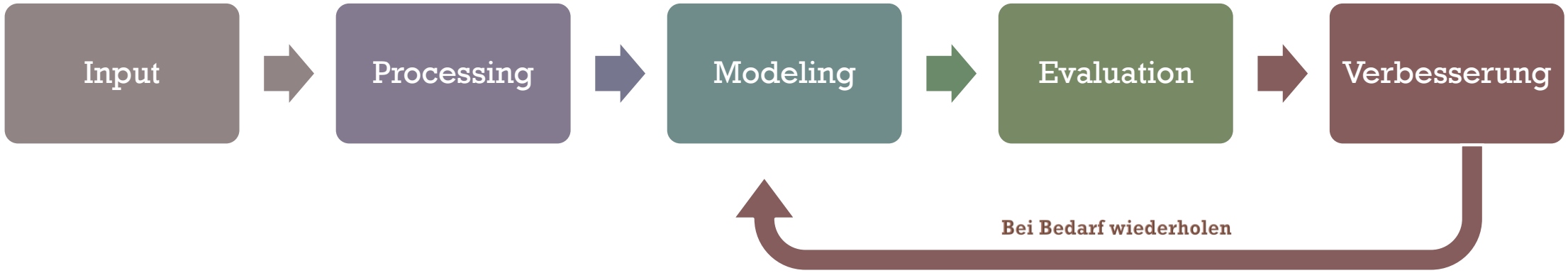
CASE STUDY

Klassifikation relevanter Apps für Life Sciences

von Iuliia G.
präsentiert am 7. August 2025

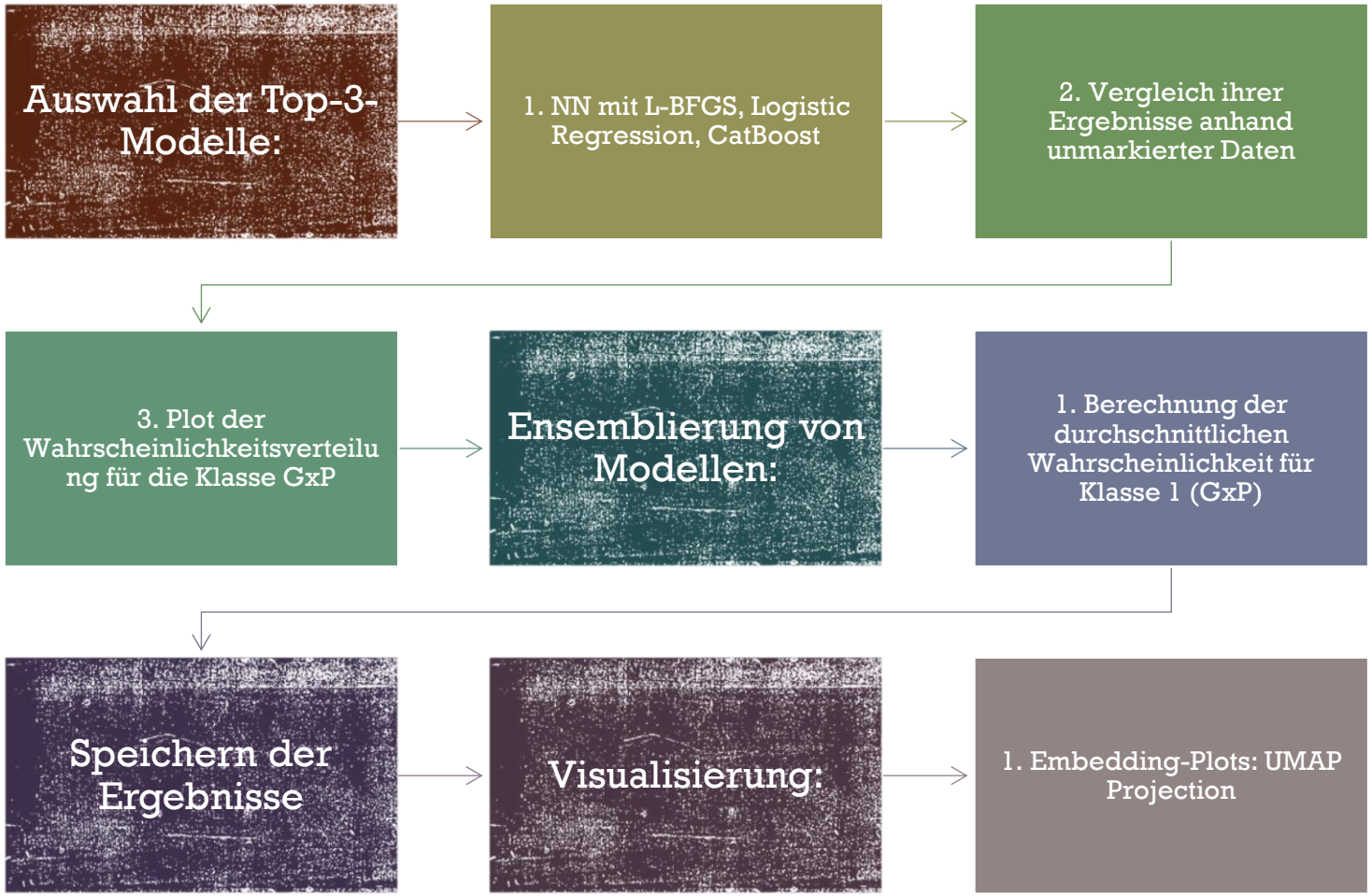
PROJEKTÜBERSICHT UND ALLGEMEINER ANSATZ

- Klassifizierung von Textbeschreibungen von Apps als GxP-relevant oder nicht relevant, ohne echte Labels im ursprünglichen Dataset



VORGEHENSWEISE (VEKTORRAUM-ANSATZ)

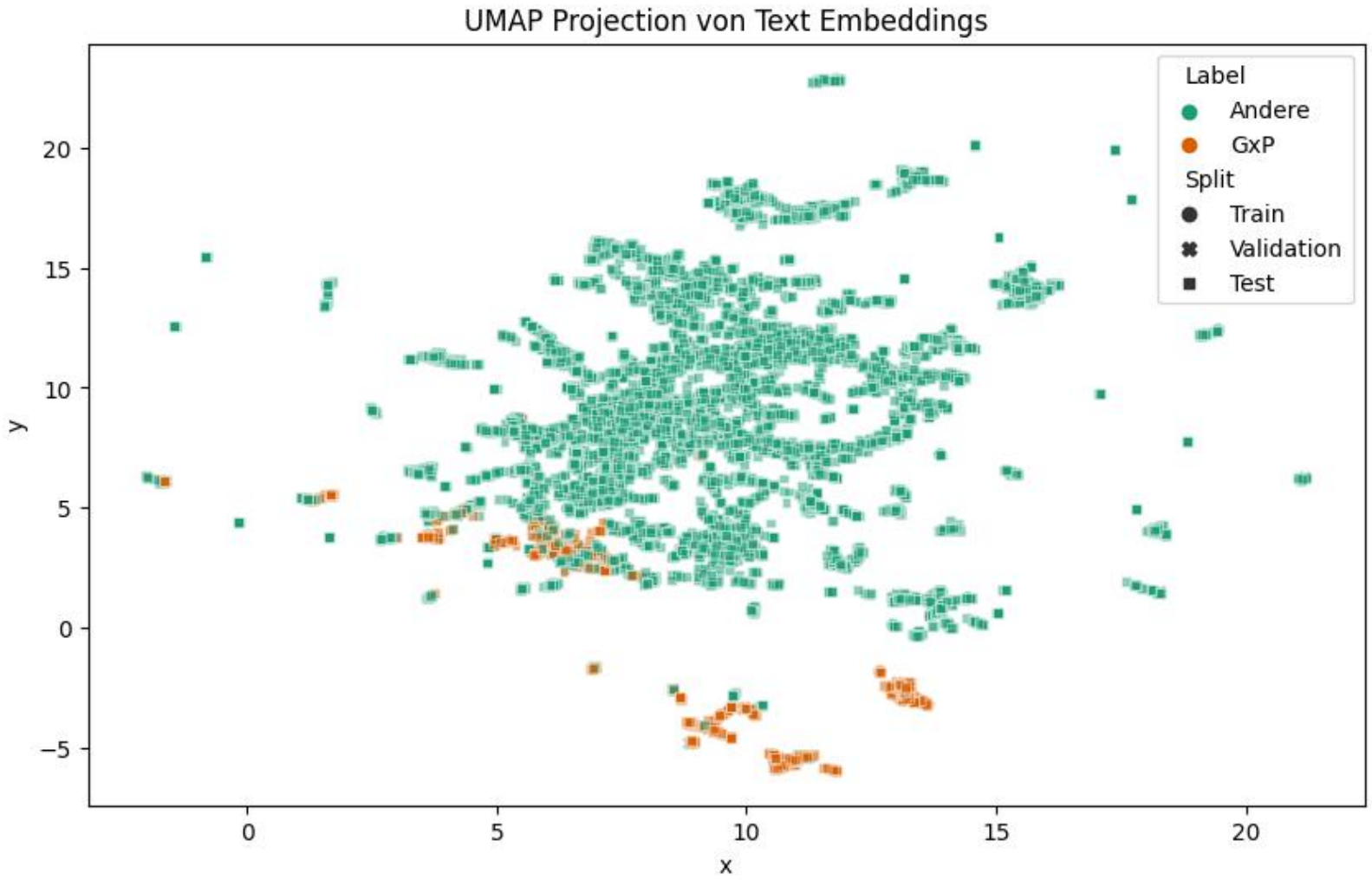
VORGEHENSWEISE (VEKTORRAUM-ANSATZ)



ERGEBNISSE & VISUALISIERUNGEN (VEKTORRAUM-ANSATZ)

Class Andere: 12935
Class GxP: 1210

(einschließlich Exemplare mit manuell zugewiesenen Labels)



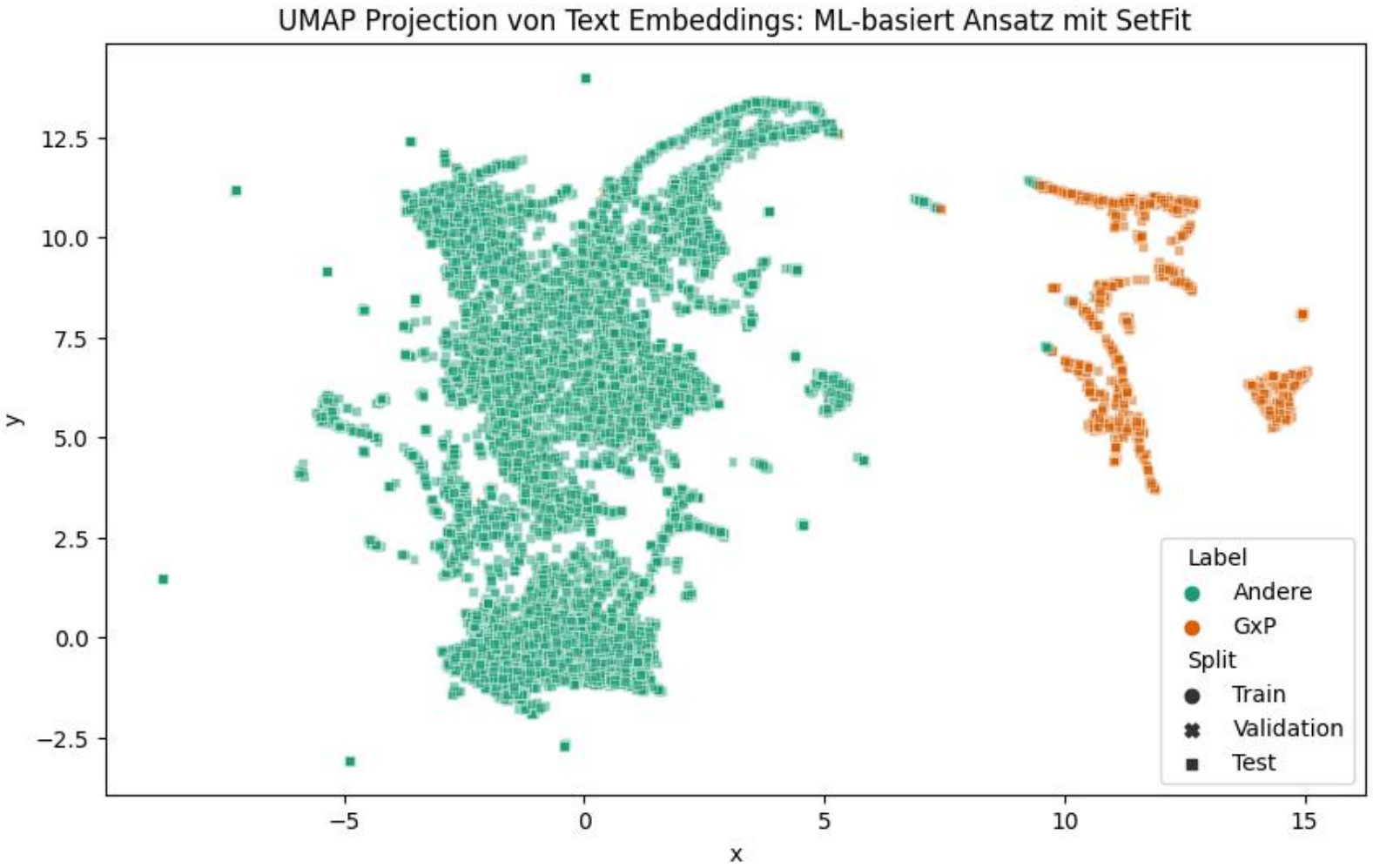
Fazit: Es gibt Bereiche, in denen die Klassen klar voneinander getrennt sind, aber es gibt auch Überschneidungen, was bedeutet, dass es für Machine-Learning-Modelle schwierig ist, eine klare Grenze zwischen den Klassen zu ziehen (wie zu sehen war). Daher sind zusätzliche Datenverarbeitung und Feature Engineering erforderlich.

VORGEHENSWEISE (ML-BASIERT - ANSATZ)

ERGEBNISSE & VISUALISIERUNGEN (ML-BASIERT - ANSATZ)

Class Andere: 12626
Class GxP: 1519

(einschließlich Exemplare mit manuell zugewiesenen Labels)

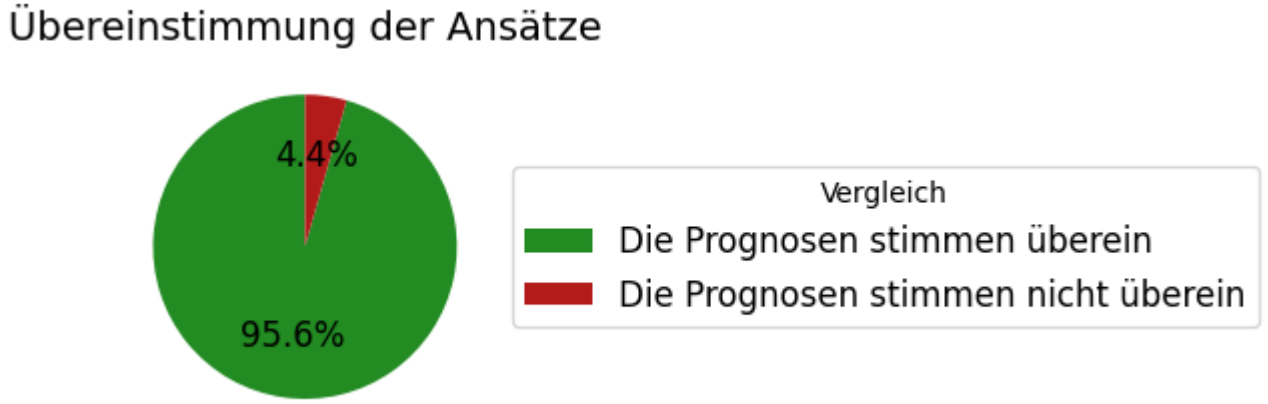


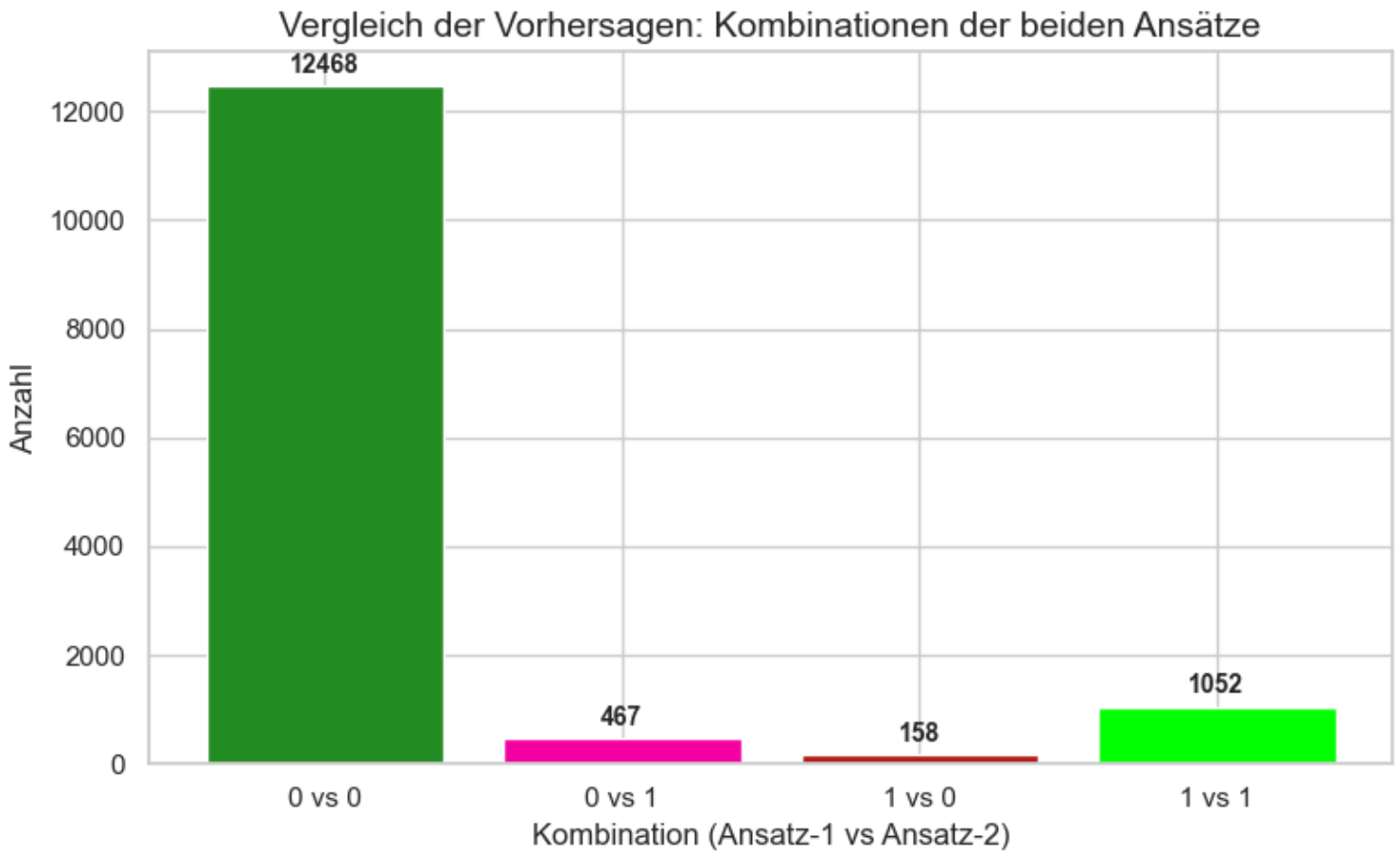
Fazit: Der ML-basierte Ansatz mit *SetFit* hat zu deutlich hochwertigeren Einbettungen mit einer klaren Klasseneinteilung geführt, bei der Überschneidungen zwischen den Klassen praktisch verschwunden sind. Texte verschiedener Klassen befinden sich nun in unterschiedlichen Bereichen des Raums, und die Aufgabe ist linear teilbar, was für Algorithmen wesentlich einfacher sein kann.

VERGLEICH DER PROGNOSEN BEIDER ANSÄTZE

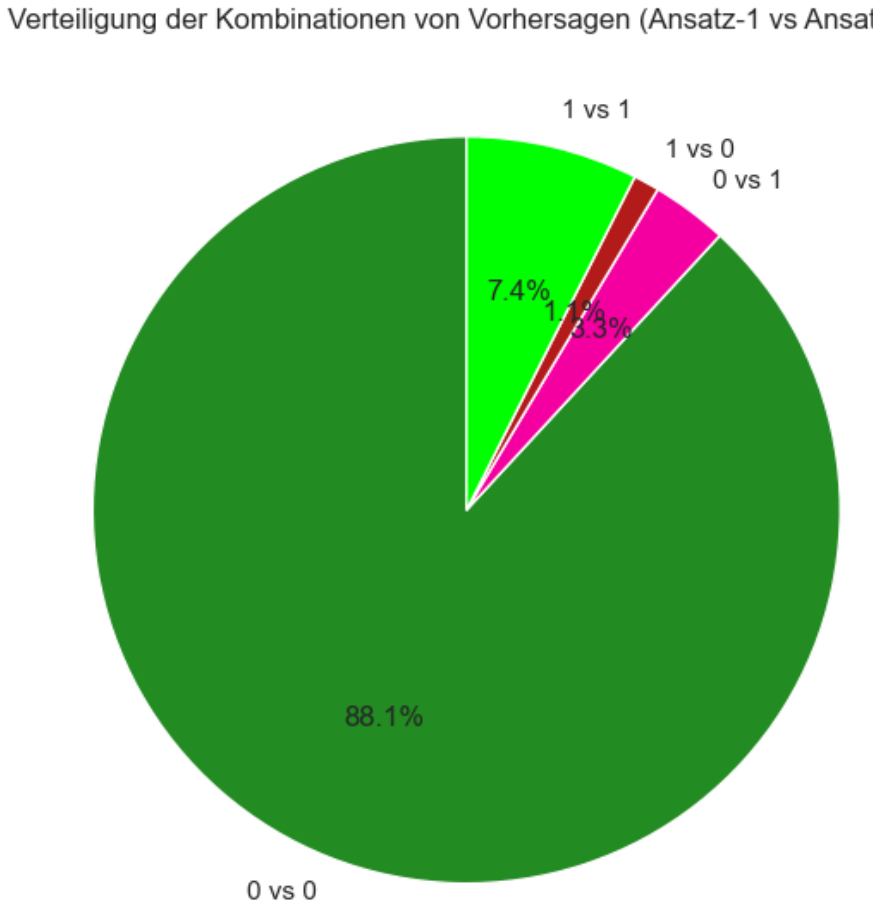
Label GxP	Ansatz-1 (Vektorraum)	Ansatz-2 (ML-basiert)
Andere (0)	12935 (91.45%)	12626 (89.26%)
GxP (1)	1210 (8.55%)	1519 (10.74%)
Total:	14145	14145

Übereinstimmen: 13520
Nicht übereinstimmen: 625





VERGLEICH DER PROGNOSEN BEIDER ANSÄTZE



VERGLEICH DER PROGNOSEN BEIDER ANSÄTZE

SCHLUSSFOLGERUNGEN

Die untersuchten Methoden erfordern weitere Untersuchungen und Experimente.

Die Unausgewogenheit der Daten beeinflusst die Prognoseergebnisse beider Ansätze – es gibt viele falsch-positive und falsch-negative Prognosen, sodass zumindest eine sorgfältigere Einstellung der Hyperparameter erforderlich ist.

Der ML-basierte Ansatz erstellt Embeddings, die viel besser in Klassen unterteilt sind.

Die Reduzierung der Anzahl Features in Ansatz 1 auf der Grundlage der Feature-Importance führte zu keinen merklichen Verbesserungen, jedoch wäre der Effekt bei einer erneuten Reduzierung möglicherweise deutlicher gewesen.

Bei Ansatz 1 ist das neuronale Netzwerkmodell mit Keras und Adam zwar flexibel, aber ohne echte Labels ist es schwierig, den Threshold und andere Parameter einzustellen. Außerdem kann es beim Neustart ohne Änderung der Parameter neue Ergebnisse liefern, was nicht mit der Reproduzierbarkeit vereinbar ist. Daher ist es nicht sehr praktisch. Die Modelle Random Forests, XGBoost und Bootstrap (Bagging) zeigten die schlechtesten Ergebnisse und sind wahrscheinlich für diesen Datentyp nicht geeignet oder erfordern eine sorgfältigere Einstellung der Parameter.

MÖGLICHE VERBESSERUNGEN (WAS ICH NOCH TUN WÜRDE)

- Das Thema GxP weiter vertiefen.
- Ich würde einen hybriden Ansatz verwenden: ML für Embeddings + Training klassischer Klassifikatoren auf diesen Embeddings.
- In Ansatz 2 würde ich alle Daten zum Trainieren des Modells verwenden, die ich markiert habe.
- Für den Ansatz 2 würde ich das vortrainierte Modell „all-mpnet-base-v2“ anstelle von „all-MiniLM-L6-v2“ verwenden.
- Ich würde die Anzahl der Epochen beim Training des Modells in Ansatz 2 erhöhen.
- Ich würde versuchen, einige weitere verwendete Funktionen (Attribute) zu entfernen – beispielsweise *Title* und *AppType*, da sie vermutlich keine nützlichen Informationen enthalten und überflüssig sein könnten.
- Ich würde versuchen, die *Feature-Importance* anders zu berechnen – über *Permutation Importance* für das bessere Modell (*LR*) und nicht über *Random Forests*.
- Auf der Grundlage der zweiten *Feature-Importance* würde ich die in Ansatz 1 verwendeten Features noch einmal überprüfen.
- Ich würde *Uncertainty Sampling* nicht nur für das Validierung-Dataset verwenden, sondern auch für das Training.
- Ich würde das *Active Learning* (in beiden Ansätzen) wiederholen.
- Ich würde *Cross-Validierung* versuchen – die Ausgangsdaten mehrfach in Train/Validation aufteilen, trainieren und die erhaltenen Metriken mitteln. Dies würde helfen, die Stabilität der Modelle zu bestimmen, ihr Überlernen zu minimieren und das beste Modell auszuwählen sowie die besten Hyperparameter, einschließlich der Regularisierung, zu finden.
- Ich würde auch andere Optionen in Betracht ziehen, wie man mit der Tendenz von Modellen zu falsch-positiven und falsch-negativen Prognosen unter den Bedingungen des bestehenden Klassenungleichgewichts umgehen kann. Zum Beispiel würde ich in der zweiten Iteration des Ansatzes 1 ebenfalls auch ein ausgewogenes Dataset erstellen.

VIELEN DANK FÜR IHRE ZEIT UND AUFMERKSAMKEIT