

Blackbox vs Whitebox Modelle für Skitourenguru

Christopher Nowzohour

Inhaltsverzeichnis

1	Motivation	1
2	Vergleich mit originalen Features	1
2.1	GAM	2
2.2	XGBoost	2
3	Benutzen von mehr Features	3
4	Fazit	5

1 Motivation

Das aktuelle Modell von Skitourenguru basiert auf einem Generalized Additive Model (GAM). Dieses ist zwar gut interpretierbar, jedoch weniger flexibel als klassische ‚Blackbox‘-Modelle wie Gradient Boosted Trees oder neuronale Netzwerke. Wir haben mit XGBoost eine leistungsfähige Blackbox-Alternative getestet und vergleichen im Folgenden deren Performance mit dem bestehenden GAM.

2 Vergleich mit originalen Features

Das aktuelle GAM benutzt nur folgende vier Features¹: `msa40`, `rd1`, `dce`, `aof`. Wir reproduzieren zuerst die Ergebnisse aus dem Paper und fitten dann ein XGBoost-Modell mit denselben Features. Die relevante Metrik ist „Area under the ROC Curve“ (AUC), welche mithilfe von cross validation (CV) oder auf einem Test-Datensatz out of sample berechnet wird. Anders

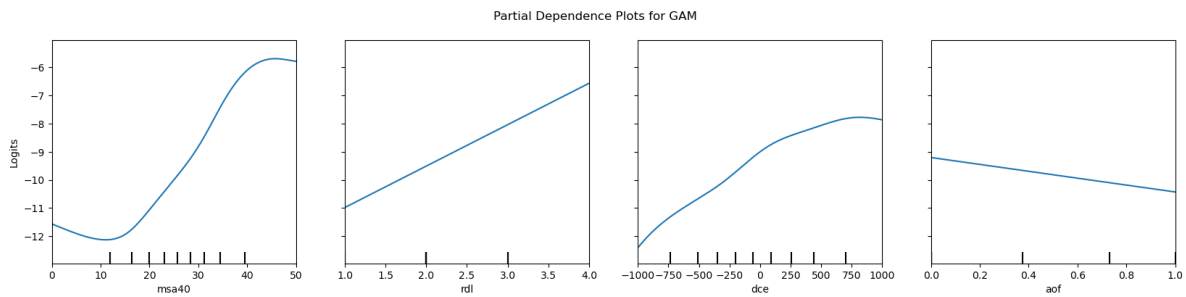
¹SLABS: An improved probabilistic method to assess the avalanche risk on backcountry ski tours.

als im Paper verwenden wir einen Test-Datensatz, d.h. wir haben ein Trainings-Datensatz mit 1000 Lawinenunfällen (auf dem wir die cross validation machen) und ein Test-Datensatz mit 250 Lawinenunfällen.

2.1 GAM

Das CV-AUC aus dem Paper beträgt 94.3. Wir kommen mit dem GAM auf ein CV-AUC von 94.0 und ein AUC auf dem Test-Datensatz von 94.7. Der kleine Unterschied ist mit den leichten Unterschieden im Evaluierungssetup zu erklären, aber für unsere Untersuchung nicht weiter relevant.

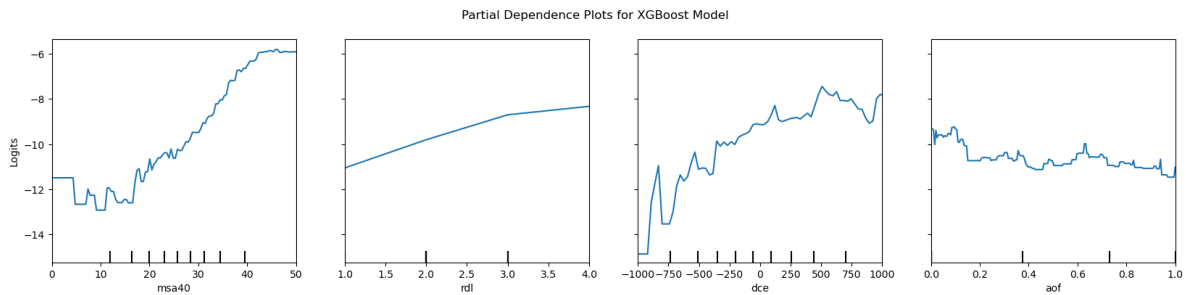
Wir können die Abhängigkeit der Modellvorhersagen von den Inputvariablen mithilfe von Partial Dependence Plots (PDPs) visualisieren.



2.2 XGBoost

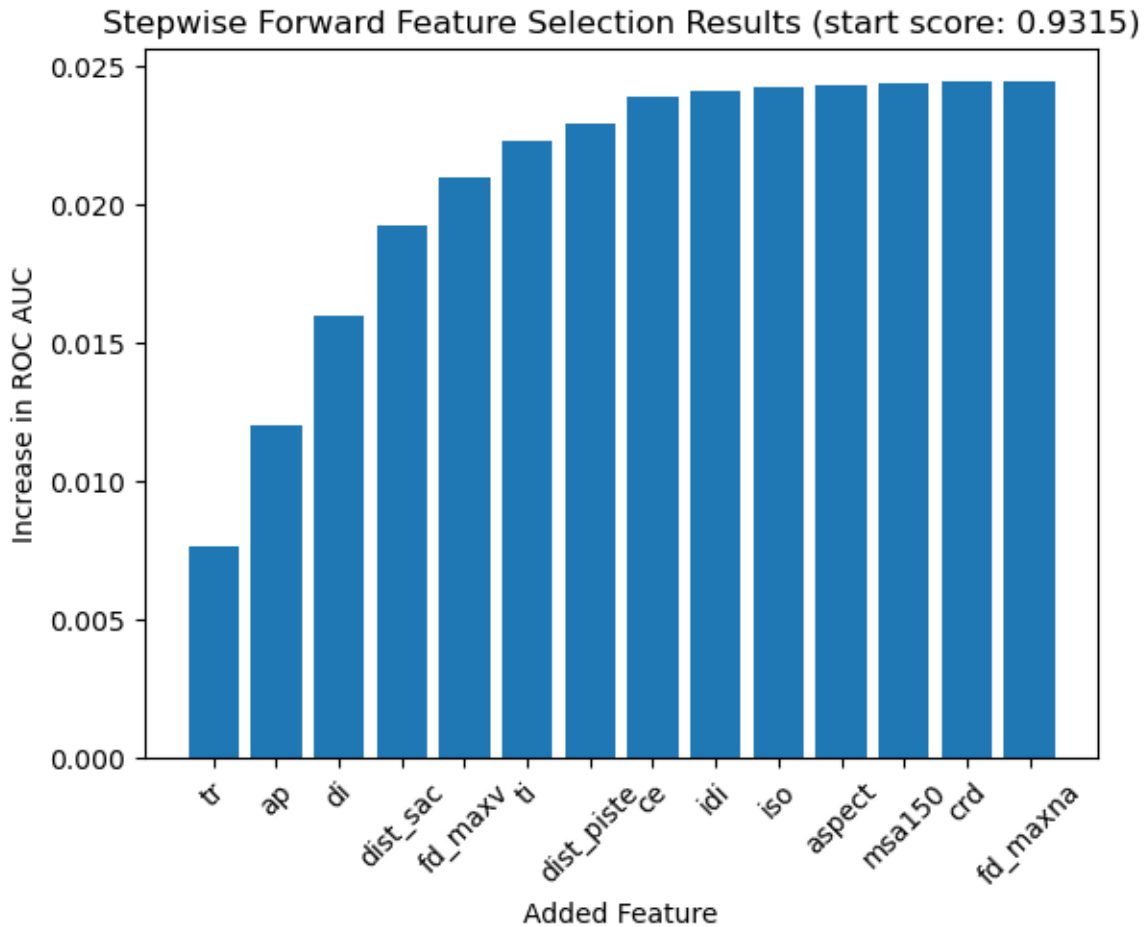
Mit XGBoost bekommen wir ein CV-AUC von 93.5 und ein AUC auf dem Test-Datensatz von 94.3.

Die PDPs für das XGBoost-Modell sind etwas weniger smooth, aber sonst denen des GAMs sehr ähnlich.



3 Benutzen von mehr Features

Im Datensatz² existieren noch viele weitere Features. Wir machen mit XGBoost eine forward selection, startend mit den vier Features im SLABS Modell, mit dem CV AUC als Zielmetrik. Wir exkludieren `sa` und `ath`, weil diese beiden Features die lokale Geländesteilheit abbilden und einen Bias ins Modell bringen können (siehe Sektion 2.3 im SLABS Paper). XGBoost scheint hier nicht zu overfitten, der CV-AUC wächst monoton mit mehr Features. Allerdings gibt es wie üblich einen abfallenden Grenznutzen.



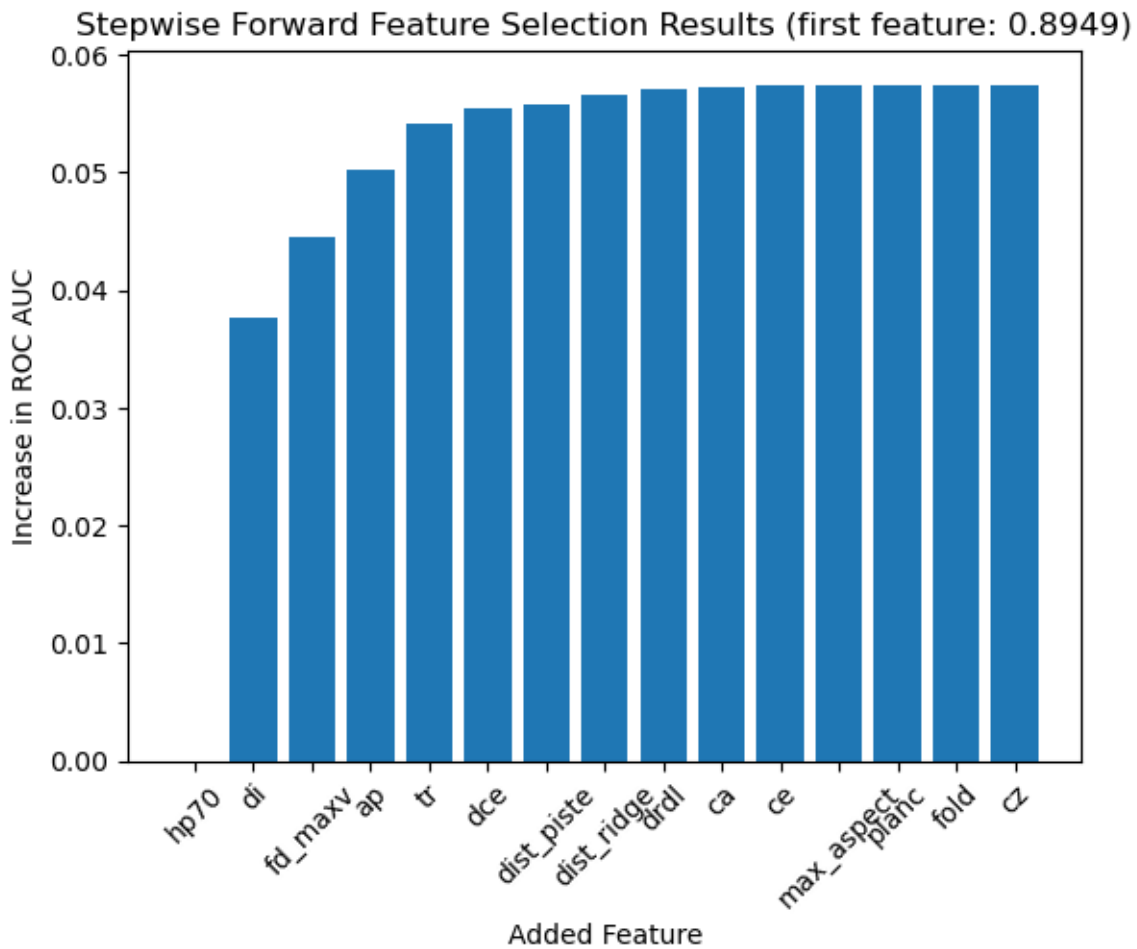
Wenn wir die ersten drei zusätzlichen Features (`tr`, `ap`, und `di`) zum Modell hinzufügen, verbessert sich sowohl mit XGBoost als auch mit GAM die Performance leicht (die Werte in Klammern sind der Unterschied zum entsprechenden Modell mit nur den vier SLABS Features).

²[Avalanche Risk Property Dataset \(ARPD\)](#)

	CV AUC	Test AUC
XGBoost	95.2 (+1.7)	95.6 (+1.3)
GAM	95.1 (+1.1)	95.8 (+1.1)

Das **ap** Feature ist speziell: es repräsentiert 7 binäre Flags, die als Bitmap kombiniert sind. Trotzdem ist das Feature auch rein numerisch interpretiert hilfreich für das Modell (hohe Werte repräsentieren zB aktive Bits vorne).

Wenn wir stattdessen die forward selection von einem leeren Modell aus starten, ergibt sich folgendes Bild (das Referenzmodell ist hier das mit dem ersten Feature).



Mit den ersten fünf Features (**hp70**, **di**, **fd_maxv**, **ap**, **tr**) bekommen wir folgende AUC-Scores.

	CV AUC	Test AUC
XGBoost	95.6 (+2.1)	95.5 (+1.2)
GAM	95.1 (+1.1)	95.6 (+0.9)

4 Fazit

Wir konnten mit XGBoost keine signifikante Mehr-Performance im Vergleich zum GAM erreichen. Das zeigt auf der einen Seite, dass das GAM die Nichtlinearitäten in den vier Features gut modelliert. Und auf der anderen Seite, dass es keine wichtigen Interaktionen zu geben scheint, die im Modell fehlen.

Unter Einbezug von mehr / anderen Features lässt sich die Performance leicht steigern, aber dafür erhöht sich auch die Modellkomplexität und es verringert sich die Interpretierbarkeit.