0.1 Vorgehensweise bei der Analyse

Bevor die Ergebnisse erläutert werden, wird kurz auf die Vorgehensweise bei der Analyse eingegangen. In die Analyse werden fünf Modelle einbezogen: SVM mit linearem, polynomialem und radialem Kern, sowie regularisierte logistische Regression und k-nearest neighbours.
Vor dem erstellen der Modelle wird ein Tuning der Hyperparameter je Modell durchgeführt. Dafür wird die Bayesian Optimization Methode genutzt, welche ein iterativer Algorithmus ist. Hierbei werden die nächsten Evaluierungspunkte basierend auf zuvor beobachteten Ergebnissen bestimmt (Yang & Shami, 2020). Der Algorithmus basiert auf zwei Hauptkomponenten: einem Surrogatmodell und einer Akqusitionsfunktion. Das Surrogatmodell, wofür hier ein Gaussian Process genutzt wird, passt die bisher beobachteten Punkte an die Zielfunktion an. Die Akquisitionsfunktion wählt dann die nächsten Punkte aus, wobei ein Gleichgewicht zwischen der Erkundung neuer Bereiche und der Nutzung vielversprechender Regionen angestrebt wird. Dafür wird hier der Ansatz des Upper-Confidence-Bound genutzt, welcher obere Konfidenzgrenzen nutzt um den Verlust gegenüber der besten möglichen Entscheidung, während der Optimierung zu minimieren (Snoek et al., 2012).

$$a_{UCB}(\mathbf{x};\ \{\mathbf{x}_n, y_n\}, \theta) = \mu(\mathbf{x};\ \{\mathbf{x}_n, y_n\}, \theta) + k\sigma(\mathbf{x};\ \{\mathbf{x}_n, y_n\}, \theta) \tag{1}$$

Die Bayesian Optimization wird genutzt, da sie eine schnelle Konvergenz für stetige Hyperparameter aufweist (Yang & Shami, 2020). Als Evaluierungskriterium wird die Genauigkeit der Modelle, welche durch den Anteil der korrekt klassifizierten Beobachtungen wiedergegeben wird, verwendet. Zur Bestimmung dieser werden die erzeugten Testdaten herangezogen.

Basierend auf den Ergebnissen des Tuning werden die oben genannten Modelle erstellt. Daraufhin werden die Genauigkeit, die Receiver Operating Characteristic Kurve (ROC-Kurve) bzw. der Area Under The Curve Wert (AUC-Wert) und der F1-Score für jedes Modell bestimmt.
Die ROC-Kurve ist eine grafische Darstellung der Leistungsfähigkeit eines Klassifikationsmodells, wobei die Sensitivität auf der y-Achse von 0 bis 1 gegen die Spezifität auf der x-Achse von 1 bis 0 abgetragen wird (Fawcett, 2006). Sensitivität und Spezifität ergeben sich aus:

$$Sensitivität = \frac{korrekt\ Positiv}{korrekt\ Positiv\ +\ falsch\ Negativ} \tag{2}$$

$$Spezifität = \frac{korrekt\ Negativ}{falsch\ Positiv\ +\ korrekt\ Negativ} \tag{3}$$

Positiv ist in diesem Fall gleichbedeutend mit Klasse 1 und Negativ mit Klasse 2. Die ROC-Kurve zeigt dann den Zusammenhang zwischen dem Nutzen (korrekt Positive) und den Kosten (falsch Positive) auf. Eine ideale Kurve läuft nah am linken, oberen Rand der Grafik, da hier bereits bei sehr hoher Spezifität (hohe Anzahl korrekt Negative) eine hohe Sensitivität (hohe Anzahl korrekt Positive) erreicht wird. Der AUC-Wert bezieht sich auf die Fläche unterhalb der Kurve und liegt somit im Intervall [0,1], wobei ein Wert von 1 für eine perfekte Klassifikation spricht, während ein Wert von 0.5 glechbedeutend mit einer rein zufälligen Zuordnung der Klassen spricht.
Für den F1-Score ist außerdem die Präzision von Bedeutung, die sich wie folgt berechnet(Fawcett, 2006):

$$Präzision = \frac{korrekt\ Positiv}{korrekt\ Positiv\ +\ falsch\ Positiv} \tag{4}$$

Der F1-Score beschreibt das harmonische Mittel zwischen Präzision und Sensitivität und drückt folglich die Fähigkeit des Modells, gleichzeitig falsch Positive und falsch Negative zu minimieren, aus.

$$F1\text{-}Score = \frac{2}{1/Präzision\ +\ 1/Sensitivität} \tag{5}$$

Zuletzt wird innerhalb jeder Datensituation, für jeden Algorithmus ein Rang vergeben, der sich aus der Summe der drei Auswertungskriterien Genauigkeit, AUC-Wert sowie F1-Score ergibt. Dabei steht Rang 1 für den am besten abschneidenden Algorithmus in der jeweiligen Datensituation.

	linear				**polynomial**				**radial**			
		ACC	AUC	F1	Rang		ACC	AUC	F1	Rang		ACC	AUC	F1	Rang
$p \ll n$	SVM-L	0.964	0.991	0.964	2	SVM-L	0.759	0.802	0.752	4	SVM-L	0.497	0.495	0.504	5
	SVM-P	0.964	0.991	0.964	2	SVM-P	0.764	0.802	0.756	2	SVM-P	0.926	0.981	0.926	1
	SVM-R	0.961	0.990	0.961	4	SVM-R	0.904	0.972	0.904	1	SVM-R	0.792	0.888	0.801	2
	LogR	0.965	0.991	0.965	1	LogR	0.762	0.802	0.751	3	LogR	0.500	0.500	0.667	4
	K-NN	0.571	0.600	0.567	5	K-NN	0.756	0.800	0.756	5	K-NN	0.687	0.749	0.657	3
		ACC	AUC	F1	Rang		ACC	AUC	F1	Rang		ACC	AUC	F1	Rang
$p \approx n$	SVM-L	0.80	0.851	0.783	3	SVM-L	0.60	0.590	0.524	3	SVM-L	0.58	0.531	0.462	5
	SVM-P	0.80	0.851	0.783	3	SVM-P	0.60	0.590	0.524	3	SVM-P	0.84	0.830	0.826	1
	SVM-R	0.86	0.885	0.851	1	SVM-R	0.60	0.586	0.545	2	SVM-R	0.74	0.889	0.794	2
	LogR	0.80	0.858	0.783	2	LogR	0.56	0.437	0.500	5	LogR	0.58	0.699	0.364	4
	K-NN	0.70	0.698	0.681	5	K-NN	0.62	0.729	0.642	1	K-NN	0.80	0.800	0.762	3
		ACC	AUC	F1	Rang		ACC	AUC	F1	Rang		ACC	AUC	F1	Rang
$p \gg n$	SVM-L	0.66	0.725	0.653	4	SVM-L	0.48	0.485	0.480	4	SVM-L	0.50	0.570	0.390	4
	SVM-P	0.70	0.750	0.681	3	SVM-P	0.50	0.517	NaN	5	SVM-P	0.76	0.726	0.750	1
	SVM-R	0.64	0.500	0.609	5	SVM-R	0.56	0.500	0.450	3	SVM-R	0.66	0.757	0.738	3
	LogR	0.72	0.754	0.741	2	LogR	0.64	0.642	0.654	2	LogR	0.54	0.538	0.343	5
	K-NN	0.80	0.866	0.808	1	K-NN	0.70	0.700	0.706	1	K-NN	0.76	0.760	0.700	2

Tabelle 1: Vergleich der Modelle

0.2 Darstellung der Ergebnisse

Tabelle 1 zeigt die Leistungsfähigkeit der fünf Klassifikationsalgorithmen über die neun verschiedenen Datensituationen anhand drei verschiedener Evaluationskriterien, sowie den Rang in der jeweiligen Datensituation. Betrachtet man den durchschnittlichen Rang über alle Szenarien hinweg, dann kann **H1** zumindest in Teilen bestätigt werden. Mit einem durchschnittl. Rang von 2.33 und 2.55 belegen jeweils *SVM-P* und *SVM-R* die vordersten Plätze. Allerdings muss auch gesagt, dass *SVM-L* nach den beiden anderen Klassifikationsmethoden den letzten Platz belegt. In den Datensituationen mit linearer Form der Entscheidungsgrenze performt insbesondere *logR* gut, da sie in allen drei Szenarien über alle Kriterien hinweg einen Wert von mindestens 0.72 aufweist und jeweils mindestens Rang 2 belegt. Im Falle von $p \ll n$ und $p \approx n$ zeigen auch *SVM-L*, *SVM-P* und *SVM-R* eine gute Leistung mit Werten nahe 0.9 bzw. 0.8. In dem hochdimensionalen Setting $p \gg n$ schneiden alle Drei, jedoch insbesondere *SVM-R* schlechter ab, welche einen AUC-Wert von 0.5 aufweist. *K-NN* belegt im linearen Kontext für $p \ll n$ und $p \approx n$ jeweils den fünften Rang, wobei der Unterschied im niedrigdimensionalen Raum am deutlichsten ist (0.39 schlechtere Genauigkeit als der nächst schlechteste Algorithmus). Im hochdimensionalen Raum kann *K-NN* allerdings als bester Algorithmus mit Werten über 0.8 überzeugen.
Die ROC-Kurven werden nur in den Datensituationen mit $p \ll n$ genutzt, da die grafische Darstellung in Szenarien mit kleinem n aufgrund der geringen Anzahl der Datenpunkte nicht sinnvoll erscheint. Des Weiteren ist es wichtig zu erwähnen, das für die Berechnung der ROC-Kurven die Wahrscheinlichkeit, dass eine Beobachtung zu einer bestimmten Klasse gehört, benötigt wird. Da SVM anders als beispielsweise logistische Regression, nicht direkt eine Wahrscheinlichkeit ausgeben, gibt es aber auch hierfür Möglichkeiten diese zu berechnen (siehe Platt, 2000).
Abbildung 1 zeigt die ROC-Kurven für die Datensituation S1. Diese zeigt grafisch noch einmal den deutlichen Unterschied zwischen *K-NN*, welcher nur etwas besser als eine reine Zufallsauswahl ist und den anderen vier Algorithmen, welche nahezu perfekt klassifizieren. Im Bezug auf **H2** sieht die Beweislage also eher dürftig aus, da *SVM-L* durschschnittlich in den linearen Szenarien eher im Mittelfeld rangiert.

In den Datensituationen mit polynomialer Form der Entscheidungsgrenze ist die Leistungsfähigkeit aller Algorithmen grundsätzlich schlechter als in den zuvor beschriebenen Szenarien. Im Fall von $p \ll n$ sind die Ergebnisse ausschließlich für *SVM-R* sehr gut mit Werten über 0.9. Die restlichen Algorithmen performen mittelmäßig mit Werten zwischen 0.7 und 0.8, wobei sich hier kein Algorithmus von den anderen absetzen kann. Bei $p \approx n$ ist *K-NN* den anderen Klassifikationsmethoden leicht überlegen, insbesondere darin gleichzeitig falsch Positive und falsch Negative zu minimieren (AUC-Wert von 0.729). Die anderen Algorithmen befinden sich durchweg über alle Werte hinweg nahe 0.5. Ähnlich ist das der Fall für $p \gg n$ bei dem auch wieder *K-NN* eine leicht bessere Leistung zeigt mit einer Genauigkeit von 0.7, gefolgt von *LogR* mit 0.64. Dennoch ist für die letzten beiden Szenarien deutlich, dass kein Algorithmus gute Leistung zeigt.
Abbildung 2 zeigt die ROC-Kurven für Szenario 2. Aus dieser Abbildung ist ersichtlich, dass sich *SVM-R* deutlich von den anderen Algorithmen abheben kann, welche alle eine mittelmäßige Leistung zeigen. Wir können also hier keine bestätigung für **H2** finden, nach der wir eigentlich erwarten würden, dass *SVM-P* hier besser Klassifiziert.

In den Datensituationen mit radialer Form der Entscheidungsgrenze ist für den Fall $p \ll n$ eine eindeutige Unterscheidung zu treffen. Während *SVM-P*, *SVM-R* und *K-NN* gut klassifizieren, haben dabei *SVM-L* und *LogR* große Probleme und zeigen lediglich Werte nahe 0.5. Insbesondere *SVM-P* zeigt mit einem AUC-Wert von 0.981 eine sehr gute Performance. Weniger drastisch aber dennoch mit dem gleichen Resultat ist dies für $p \approx n$ der Fall. Während *SVM-P* im Vergleich zum vorherigen Szenario etwas schlechter performt, bleibt *SVM-R* nahezu identisch und *K-NN* kann sich sogar leicht steigern auf einen AUC-Wert von 0.8. *SVM-L* und *LogR* können sich leicht verbessern (beispielsweise mit einer Genauigkeit nahe 0.6), belegen dennoch weiterhin Rang vier und fünf. Für $p \gg n$ sind ebenfalls die beiden Gruppen zu differenzieren. Für *SVM-L* und *LogR* sind wie in Szenario 3 wieder Werte nahe 0.5 für die Genauigkeit erkennbar. *SVM-P* und *K-NN* performen in diesem Szenario am Besten. *SVM-R* fällt insbesondere mit einer Genauigkeit von 0.66 im Vergleich zu 0.76 etwas hinter den anderen beiden Algorithmen zurück.
Abbildung 3 zeigt die ROC-Kurven für Szenario 3. Hieraus wird deutlich, dass die Kurve von *SVM-P* nahe an der oberen, linken Ecke verläuft, was für eine nahezu perfekte Klassifikation spricht. Der etwas flachere Verlauf von *SVM-R* und *K-NN* macht deutlich, dass diese beiden Algorithmen weniger gute Leistung zeigen.

Dennoch können die beiden oben erwähnten Gruppen gut identifiziert werden, da sich *SVM-L* und *LogR* nahe der diagonalen Linie orientieren. Diese beschreibt eine ROC-Kurve einer reinen Zufallsauswahl, woraus gefolgert werden kann, dass sie kaum Fähigkeit aufweisen zwischen den Klassen zu unterscheiden. Diese Szenarien bestätigen die Vermutung aus **H2** zwar nicht direkt, da hier im Ranking über alle Szenarien mir radialer Entscheidungsgrenze hinweg interessanterweise *SVM-P* am besten performt aber zumindest liegt hier *SVM-R* auf dem zweiten Platz. Trotzdem müssen wir mit diesen Ergebnissen **H2** ablehnen.

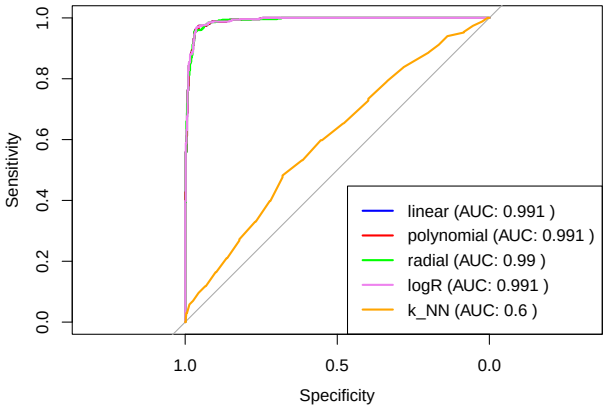

Abbildung 1: ROC-Kurven Szenario 1

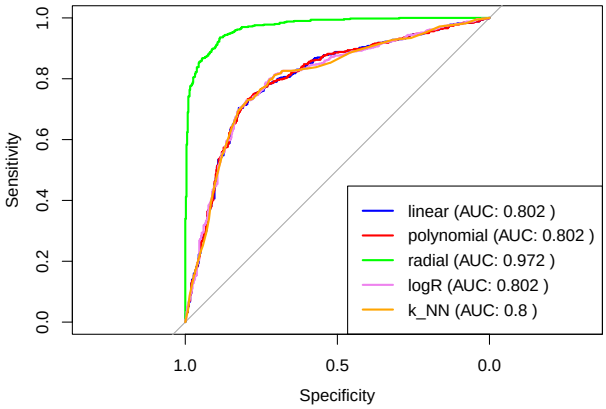

Abbildung 2: ROC-Kurven Szenario 2

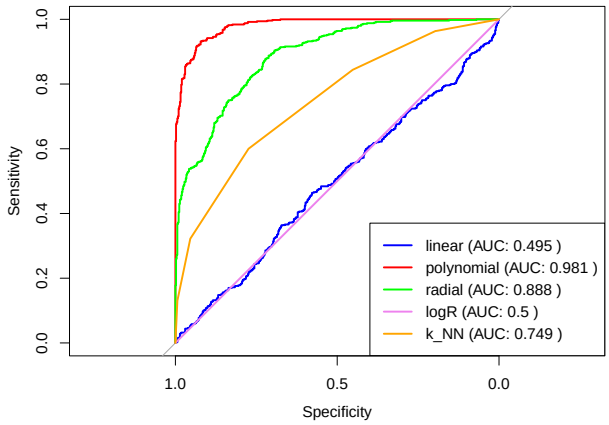

Abbildung 3: ROC-Kurven Szenario 3

Zuletzt wird ein Vergleich der einzelnen Dimensionierungen über die drei Formen der Entscheidungsgrenze gezogen. So ist für $p \ll n$ ersichtlich, dass insbesondere *SVM-P* und *SVM-R* über alle drei Szenarien gute Leistung (mit Werten mindestens nahe 0.8). Auch im durchschnittlichen Ranking für diese Szenarien über alle drei Szenarien zeigen die *SVM*-Methoden die besten Ergebnisse. Überaschen ist nur die unerwartet schlechte Performance von *K-NN*. Somit bestätigt sich **H3** hier nur teilweise. In den hochdimensionalen Szenarien $p \gg n$ ist einzig *K-NN* überzeugend, welcher auch im polynomialen Szenario Werte über 0.7 aufweist. Dies deutet klar darauf hin, dass **H4** mit unseren Daten verworfen werden muss.

Literatur

Fawcett, T. (2006). An Introduction to ROC Analysis. *Pattern Recognition Letters*, *27*(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Platt, J. (2000). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Adv. Large Margin Classif.*, *10*.

Snoek, J., Larochelle, H., & Adams, R. P. (2012, 29. August). *Practical Bayesian Optimization of Machine Learning Algorithms*. arXiv: 1206.2944 [cs, stat]. Zugriff 19. August 2024 unter http://arxiv.org/abs/1206.2944

Yang, L., & Shami, A. (2020). On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice. *Neurocomputing*, *415*, 295–316. https://doi.org/10.1016/j.neucom.2020.07.061