Das Ziel dieser Arbeit ist es, in verschiedenen Datensituationen die Performance von SVM Algorithmen für die binäre Klassifikation zu evaluieren. Dafür wollen wir eine Reihe von Datensätzen mit verschiedenen Charakteristiken synthetisch erstellen und entsprechend als Trainingsdaten verwenden. Die Datensätze unterscheiden sich in zwei zentralen Eigenschaften. Das erste sind die Dimensionen. Es soll unterschieden werden in drei Kategorien. Die erste ist, dass es deutlich mehr Beobachtungen als Variablen gibt, also $n \gg p$. Ein solches Datenszenario kann z.B. im Kontext des Zensus auftreten, in dem eine große Anzahl an Personen befragt wird, aber die Vorgabe besteht, dass die Bürger nicht zu stark belastet werden sollen, weshalb nur einige wenige Kernfragen gestellt werden. In diesem Fall wurde für $n = 1000$ Beobachtungen und $p = 10$ Variablen generiert. Das zweite Szenario stellt Datensätze vor die etwa gleich viele Beobachtung wie Variablen haben $n \approx p$. Ein solches Szenario kann in vielen Kontexten auftreten. Für dieses Szenario sind die Dimensionen $n = p = 50$. Das letzte Szenario behandelt dann entsprechend den Fall $n \ll p$. Dies tritt oft im Kontext von Datenerhebungen im medizinischen Bereich auf, da sehr viele Erhebungen zu kostspielig wären. Auch im Bereich des Natural Language Processing sind solche Datensätze häufiger anzutreffen (Scholz & Wimmer, 2021). Die Werte die dafür angenommen wurden sind $p = 200$ und $n = 50$.

Die zweite Charakteristik ist der Datengenerierende Prozess. Da in dieser Arbeit SVM im Vordergrund stehen und wir hier vor allem zeigen wollen, wie SVM funktionieren, wird der DGP so aufgebaut, dass er der grundlegenden Idee der SVM am ehesten entspricht. Die Vorgehensweise ist, im ersten Schritt eine Hyperplane, im p-Dimensionalen Raum, in einer bestimmte Form zu erstellen und anschließend auf jeweils einer Seite dieser Hyperplane $n/2$ zufällige Punkte zu samplen, die die jeweilige Ausprägung in der Zielvariable repräsentieren. Wir haben also für die Zielvariable eine 50/50 Verteilung für die binären Ausprägungen angenommen.

Insgesamt gibt es auch hier wieder 3 Kategorien. Die erste sind linear getrennte Daten. Dafür wird eine lineare Hyperplane der Form

$$\beta_1 X_1 + \beta_2 X_2 + ... + \beta_{1-p} = \beta_p X_p$$

mit zufälligen Koeffizienten erzeugt. Diese Daten sollen also den Annahmen entsprechend, die für SVMs mit linearem Kernel gelten. Nachem für eine Beobachtung j zufällig ein Punkt auf der Ebene gesampelt wurde, wurde dieser anschließend verschoben. Die Verschiebung erfolgte über eine Skalierung des normierten Normalenvektors $k\left(\frac{\overline{\beta}}{||\overline{\beta}||}\right)$. k ist dabei eine normalverteilte Zufallszahl mit Mittelwert μ_k und Varianz σ_k^2. Dieser Prozess wird $n/2$ mal wiederholt für die eine Ausprägung der Zielvariable ($y_i = 1$) und dementsprechend $n/2$ mal für die andere Ausprägung ($y_i = -1$), dann aber mit $-\mu_k$ als Mittelwert für k.
In der zweiten Situation hat die Hyperplane eine quadratische Form:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2 + ... + \beta_{2p-2} X_{p-1} + \beta_{2p-1} X_{p-1}^2 = \beta_{2p} X_p$$

diese Form der Trennung stellt also eine Merkmalserweiterung um quadratische Terme dar und funktioniert damit ähnlich wie eine SVM mit polynomialen Kernel mit $d = 2$. Die Verschiebung erfolgte hier nur durch Skalierung der Werte von X_p ebenfalls mit $k \sim \mathcal{N}(\mu_k, \sigma_k^2)$ für die eine Ausprägung und $k \sim \mathcal{N}(-\mu_k, \sigma^2)$ für die andere Ausprägung.
Der letzte DGP geht von einer noch komplexeren Entscheidungsgrenzen aus. Es wird hier ein Hypershpäre im p dimensionalen Raum erstellt und einmal innerhalb und einmal außerhalb dieser gesampelt. Dafür wurde für eine Beobachtung j, $p-1$ Winkel θ zufällig erstellt, ein Radius r festgelegt und anschließend die einzelnen Werte $X_{1,j}, X_{2,j}, ..., X_{p,j}$ berechnet. Die Berechnung erfolgt dabei über die Definition von sphärischen Koordinaten:

$$\begin{aligned}
X_{1,j} &= r\cos(\theta_1) \\
X_{2,j} &= r\sin(\theta_1)\cos(\theta_2) \\
X_{3,j} &= r\sin(\theta_1)\sin(\theta_2)\cos(\theta_3) \\
&\vdots \\
X_{p-1,j} &= r\sin(\theta_1)\ldots\sin(\theta_{p-2})\cos(\theta_{p-1}) \\
X_{p,j} &= r\sin(\theta_1)\ldots\sin(\theta_{p-2})\sin(\theta_{p-1})
\end{aligned}$$

Dieser Vorgang wird dann $n/2$ mal, wiederholt und anschließend erfolgt die Verschiebung durch eine Skalierung des jeweiligen normalisierten Datenvektors $k\left(\frac{\overline{x}}{||x||}\right)$. k ist auch hier wieder eine Zufallsvariable mit μ_k für die eine Ausprägung der Zielvariable und $-\mu_k$ für die andere Ausprägung. Die Streuung bleibt bei beiden bei einem konstanten Wert σ_k^2. Je nachdem wie μ_k und σ_r^2 gewählt werden kann die Trennbarkeit der Daten angepasst werden. Für alle Szenarien wurde ein Wert für μ_k und σ_k^2 festgelegt, der für eine moderate Überschneidung der zwei Klassen sorgt. Allerdings musste der Abstand in den höherdimensionalen Szenarien etwas erhöht werden, da sonst die Performance für alle Classifier sehr schlecht und damit nicht vergleichbar war.

Es ergeben sich daher 9 unterschiedliche Datensituationen, welche in ihren Dimensionen und Komplexität der Entscheidungsgrenze variieren. Die Kürzel für die Situationen sind in Tabelle 1 abgetragen

	linear	polynomial	radial
$p \ll n$	S1	S2	S3
$p \approx n$	S4	S5	S6
$p \gg n$	S7	S8	S9

Tabelle 1: Datenszenarien

Zusätzlich soll nicht nur ein Vergleich zwischen der Performance der SVM mit verschiedenen Kernels gemacht werden, sondern auch die Unterschiede in der Klassifikationsgüte der SVM zu anderen gängigen Klassifikationsmethoden gezeigt werden. Dafür werden regularized Logistic Regression und k-nearest Neighbour als Vergleichsalgorithmen hinzugezogen.

Literatur

Scholz, M., & Wimmer, T. (2021). A Comparison of Classification Methods across Different Data Complexity Scenarios and Datasets. *Expert Systems with Applications*, *168*, 114217. https://doi.org/10.1016/j.eswa.2020.114217