Frederik, Ludwigs:

Einfluss von Hyperparametern bei Support Vector Machines

Bachelor, Winter Semester 2017

Faculty of Mathematics, Computer Science and Statistics
UNSPECIFIED
UNSPECIFIED

Ludwig-Maximilians-Universität München

https://doi.org/10.5282/ubm/epub.38414

Ludwig-Maximilians-Universität
München

Institut für Statistik

Bachelorarbeit

Einfluss von Hyperparametern bei Support Vector Machines

Autor:
Frederik
Ludwigs

Betreuung:
Philipp Probst
Prof. Dr. Anne-Laure
Boulesteix

23. März 2017

Abstract

Diese Arbeit beschäftigt sich mit der Methode der Support Vector Machines. Dabei handelt es sich sowohl um eine Regressions- als auch um eine Klassifizierungsmethode, wobei diese Arbeit nur die Klassifizierungsmethode behandelt. Diese besitzt dabei verschiedene Hyperparameter - Einstellungsmöglichkeiten für die Methode - die teilweise einen sehr starken Einfluss auf die Modellgüte haben. Ziel dieser Arbeit ist es, anhand von 128 realen Datensätze die verschiedenen Einstellungen für diese Hyperparameter zu untersuchen und inwieweit diese die Modellgüte beeinflussen. Die Güte der Modelle wird dabei anhand von vier verschiedenen Gütemaßen und dem Verfahren der Kreuzvalidierung gemessen.
Zuerst werden die theoretischen Grundlagen für das Verfahren der Kreuzvalidierung vorgestellt und anschließend die Theorie der verschiedenen Gütemaße erläutert. Bezüglich der Support Vector Machines werden erst deren Grundlagen dargestellt und veranschaulicht. Anschließend wird die Theorie der Support Vector Machines erläutert sowie deren verschiedene Hyperparameter vorgestellt. Der Einfluss der Hyperparameter auf die Methode der Support Vector Machines wird dabei anhand von simulierten Daten veranschaulicht. Im Hauptteil wird mit Hilfe von 128 Datensätzen der Einfluss der Hyperparameter auf die Modellgüte untersucht. Dabei gibt es verschiedene Tendenzen zu beobachten. Werden manche der Hyperparameter zu hoch gesetzt, so kann dies dazu führen, dass sich im Mittel die Modellgüte für die 128 Datensätze verschlechtert, sodass sich sagen lässt, dass diese Hyperparametereinstellungen bei dem Großteil der Datensätze eher ungeeignet ist. Auch ist die durchschnittliche Güte der Modelle bei bestimmten Einstellungen für die Hyperparameter deutlich besser als bei anderen Einstellungen, da bei diesen Einstellungen besonders oft zufriedenstellende Ergebnisse resultieren.

Inhaltsverzeichnis

1 Einleitung

Die Datenmenge, die heutzutage täglich erzeugt wird, ist im Vergleich zu der Zeit vor 15 Jahren schier unendlich. Alleine Social Media Plattformen wie Facebook, Instagram, Twitter und Youtube generieren dabei gigantische Mengen. So wird nur auf der Videoplattform Youtube jede Minute vier Stunden Videomaterial hochgeladen. Auf Instagram werden minütlich 2.5 Millionen Bilder geliked, auf Twitter täglich 500 Millionen Tweets geteilt und auf Facebook jede Woche über 30 Milliarden Nachrichten verschickt. Das sind die Datenmengen, die alleine durch wenige Social Media Plattformen generiert werden. Neben diesen gibt es auch noch andere Social Media Plattformen und Internetseiten, so wie beispielsweise Google, das jeden Tag etwa sechs Milliarden Suchanfragen verarbeitet [vgl. [Gwava, 2016]]. Doch auch außerhalb des gewöhnlichen Internets werden jeden Tag enorme Datenmengen erzeugt. Beispielsweise laden Connected Cars, also Autos, die mit dem Internet verbunden sind, pro Stunde etwa 25 Gigabyte an Daten, wie z.B. Ort, Geschwindigkeit, Straßenbedingungen etc., ins Internet [vgl. [Hitachi, 2016]]. Marktbeobachter gehen davon aus, dass die Menge an Daten, die entweder erstellt, vervielfältigt oder konsumiert wird, bis zum Jahr 2020 bei etwa 40 Zettabyte, was knapp 40 Millionen Terabyte entspricht, liegen wird [vgl. [Jüngling, 2013]]. Dementsprechend sind in immer größer werdenden Mengen Daten vorhanden, was dazu führt, dass Methoden des statistischen Lernens in vielen Bereichen, wie z.B. in der Forschung, der Medizin, im Marketing und in der Finanzwelt, eine immer wichtigere Rolle spielen [vgl. [Witten et al., 2013] S. VII]. Dabei bezeichnet der Begriff des statistischen Lernens verschiedene statistische Modelle, wie beispielsweise Klassifikations- und Regressionsmodelle. Hierbei wird zwischen überwachten und unüberwachten Lernmethoden unterschieden. Bei überwachten Methoden handelt es sich um solche, die zu jeder Beobachtung eine gewisse Anzahl an Einflussvariablen und einen Responsewert benötigen. Bei unüberwachten Methoden hingegen werden nur eine gewisse Anzahl an Einflussvariablen gebraucht und kein Responsewert. Bei den Methoden des statistischen Lernens spielen häufig sogenannte Hyperparameter - Einstellungsmöglichkeiten für solche Methoden - eine nicht unwichtige Rolle, da diese die Qualität der Methoden stark beeinflussen können. Ziel dieser Arbeit ist es anhand von realen, also nicht simulierten Datensätzen den Einfluss verschiedener Hyperparameter für die statistische Lernmethode Support Vector Machines zu untersuchen. Zuerst wird die Theorie zu Support Vector Machines sowie zur Schätzung der Genauigkeit einer Methode erläutert. Anschließend soll der Einfluss von verschiedenen Einstellungen bei Hyperparametern untersucht werden.

2 Beurteilung der Güte von Modellen

Dieses Kapitel behandelt Methoden und Maße zur Bewertung der Güte von statistischen Modellen. Dies schließt sowohl die Unterscheidungen von verschiedenen Fehlerarten als auch die theoretischen Grundlagen für Resampling-Verfahren ein. Zudem werden verschiedene Gütemaße vorgestellt.

2.1 Trainigs- und Testfehler

Bei dem Erstellen von statistischen Modellen ist es fundamental wichtig zu schätzen, wie genau die Prognose eines Modelles sein wird. Im allgemeinen bezeichnet dabei y_i den wahren Responsewert und $\hat{y}_i$ den durch das Modell prognostizierten Wert. Bei Klassifikationsverfahren bezeichnen y_i und $\hat{y}_i$ Klassenzugehörigkeiten, und die Fehlerrate, die das Modell macht, wird durch

$$\theta = \frac{1}{n} \sum_{i=1}^{n} I(y_i \neq \hat{y}_i) \tag{1}$$

bestimmt, wobei n die Anzahl der Beobachtungen darstellt [Witten et al., 2013][S. 37]. $I(y_i \neq \hat{y}_i)$ bezeichnet eine Indikatorfunktion, welche 1 ist, falls $y_i \neq \hat{y}_i$ gilt, beziehungsweise 0, falls $y_i = \hat{y}_i$ gilt. Es gilt $\theta \in [0, 1]$, dabei gilt je kleiner θ, umso geringer die Fehlklassifizierungsrate des Modells.
Es gibt eine wichtige Unterscheidung von Fehlertypen. Auf der einen Seite der Trainigsfehler, der die Fehlerrate θ_{Train} bezeichnet, die man erhält, wenn man θ auf Basis der Daten berechnet, mit denen das Modell erstellt wurde. Auf der anderen Seite der Testfehler, der die Fehlerrate θ_{Test} angibt, die durch das Modell gemacht wird, wenn man dem Modell neue Daten übergibt, also solche, die nicht zur Modellierung verwendet wurden. Normalerweise ist der Testfehler relevanter als der Trainingsfehler, da man sich im allgemeinen eher dafür interessiert, wie genau die Prognose des Modells sein wird, wenn man ihm neue Daten übergibt. Fitten wir also ein Modell auf Basis von n Beobachtungen, ist es interessanter zu wissen, wie genau die Prognose des Modells mit neuen Datenpunkten ist als zu wissen, wie genau die Prognose mit den bereits gesehen n Datenpunkten ist. Dementsprechend ist man auch an dem Modell interessiert, das die geringste Testfehlerrate θ_{Test} besitzt, da dessen Prognosen mit neuen Daten am genausten sein wird. Dass das Modell mit dem geringsten Trainingsfehler nicht zwangsläufig das Modell mit dem geringsten Testfehler sein muss, soll nun folgendes simuliertes Beispiel, das sich an [Witten et al., 2013][S. 361] anlehnt, verdeutlichen.

Es wurden insgesamt 2 mal 200 standardnormalverteilte Datenpunkte erzeugt, von denen die ersten 150 Punkte jeweils mit 2 addiert bzw. subtrahiert wurden. Diese Datenpunkte fallen in Klasse $y = -1$, die restlichen Datenpunkte in Klasse $y = 1$. Der Scatterplot der simulierten Daten ist in Abbildung (1) dargestellt.

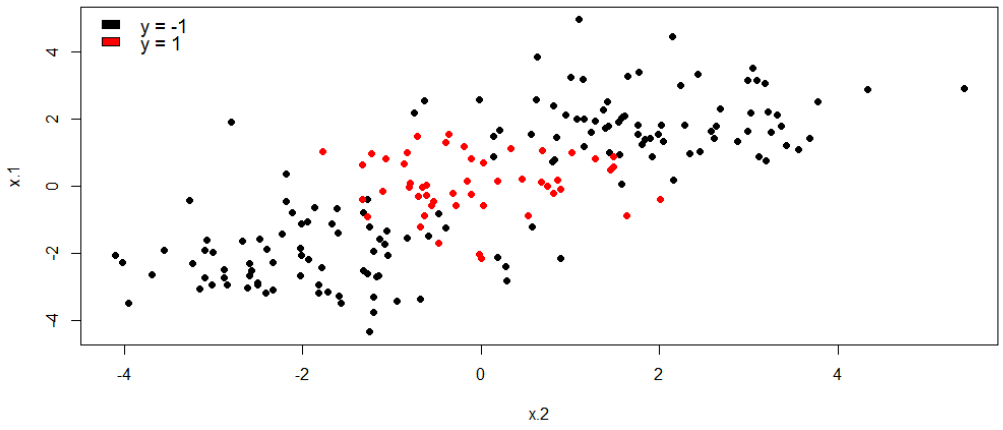


Abbildung 1: 200 simulierte Datenpunkte mit zwei verschiedenen Klassenzugehörigkeiten

Im nächsten Schritt wurden die simulierten Daten in zwei Teildatensätze mit jeweils 100 zufällig ausgewählten Beobachtungen aufgeteilt. Dabei dient einer dieser Teildatensätze als Trainingsdatensatz, auf dessen Basis das Modell erstellt wird, und der zweite Teildatensatz wird zur Bestimmung von θ_{Test} verwendet und ist dementsprechend ein Testdatensatz. Anhand des Trainigsdatensatzes wurden zwei verschiedene Modelle erstellt, die sich nur bezüglich eines Hyperparameters unterscheiden. Beide Modelle sind in Abbildung (2) dargestellt. Datenpunkte, die in den oliven Bereich fallen, werden der Klasse $y = 1$ zugeordnet, Datenpunkte, die in den grauen Bereich fallen, der Klasse $y = -1$. Die in rot dargestellten Trainingspunkte sind die Daten, die zur Klasse $y = 1$ gehören und die in schwarz dargestellten Trainingspunkte diejenigen, für die $y = -1$ gilt. Schwarze Datenpunkte im oliven Bereich bzw. rote Datenpunkte im grauen Bereich sind also fehlklassifizierte Datenpunkte. Das erste Modell - links in Abbildung (2) - ist bereit einige Fehlklassifizierungen einzugehen, anders als das zweite Modell, - rechts in Abbildung (2) - das extrem stark versucht Fehlklassifizierungen zu vermeiden. Dementsprechend sind die Bereiche des zweiten Modells wesentlich stärker an einzelnen Datenpunkten orientiert als beim ersten Modell.

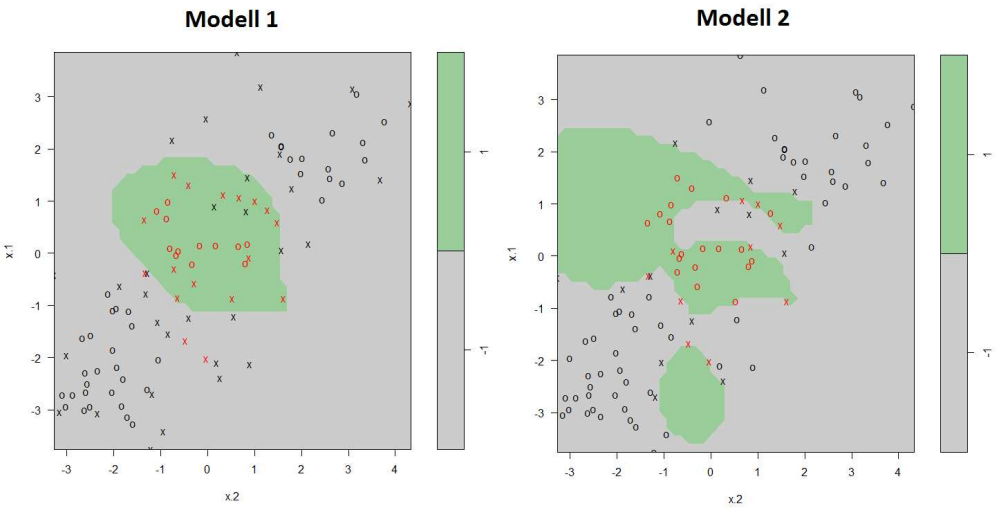


Abbildung 2: Links: Modell 1 mit geringem restriktivem Hyperparameter Rechts: Modell 2 mit extrem restriktivem Hyperparameter

Das erste Modell mit weniger striktem Hyperparameter klassifiziert insgesamt acht (Trainigs-)Datenpunkte falsch und hat dementsprechend eine Fehlerrate θ_{Train} von 8%. Das zweite Modell mit dem wesentlich strikteren Hyperparameter macht insgesamt nur eine Fehlklassifizierung und hat folglich eine Fehlerrate θ_{Train} von 1%. Man könnte also meinen, dass das zweite Modell wesentlich besser geeignet sei, um die Klassenzugehörigkeit anhand der Variablen $x.1$ und $x.2$ zu prognostizieren. Nimmt man nun allerdings den zweiten Teildatensatz, der nicht zur Berechnung des Modells verwendet wurde, und berechnet damit den Testfehler θ_{Test} für die beiden Modelle, so liegt der Testfehler θ_{Test} für das erste Modell mit 12% weit unter dem Testfehler θ_{Test} des zweiten Modells, der sich bei 20% befindet. Es wird also ersichtlich, dass das erste Modell besser geeignet zu sein scheint, um neue Werte zu prognostizieren als das zweite Modell. Dementsprechend würde man sich für das erste Modell entscheiden. Genau das Gegenteil würde man behaupten, hätte man nur den Trainingsfehler berücksichtigt, da dieser beim zweiten Modell weit unter dem des ersten Modells liegt. Dieses Problem lässt sich darauf zurück führen, dass sich das zweite Modell zu sehr an einzelnen Datenpunkten orientiert; dieses Probelm wird auch *Overfitting*, zu deutsch 'Überanpassung', genannt.

2.2 Schätzung des Testfehlers

Wie das Beispiel aus Kapitel 2.1 gezeigt hat, können sich Test- und Trainingsfehler stark voneinander unterscheiden. Dabei hat der Testfehler eine wesentlich höhere Aussagekraft als der Trainingsfehler. So lässt sich die Verwendung eines Modelles durch einen geringen Testfehler rechtfertigen, da das Modell auch unbekannte Werte gut zu prognostizieren/ klassifizieren weiß. Es kommt allerdings nicht sonderlich oft vor, dass man zwei seperate Datensätze hat, um so Modell und Testfehler getrennt von einander zu berechnen. Es gibt allerdings verschiedene Methoden, um den Testfehler zu schätzen, auch wenn man eigentlich nur einen Datensatz hat. Bei diesen Methoden wird der Datensatz in Test- und Trainingsdatensatz aufgeteilt, wobei letzterer zur Berechnung des Modells verwendet wird und der Testdatensatz zur Schätzung des Testfehlers. Dazu gibt es verschiedene Ansätze.
Bei dem *Validierungsdatensatz Ansatz* wird der Datensatz zufällig in zwei etwa gleich große Datensätze aufgeteilt. Dabei dient einer dieser (Teil-)Datensätze zur Modellierung und der andere zur Berechnung des Testfehlers (entspricht dem Vorgehen des Beispiels in Kapitel 2.1). Dieses recht einfache Vorgehen hat allerdings auch einige entscheidende Nachteile. Da die Daten zufällig in zwei Teildatensätze aufgeteilt werden, ist die Schätzung für θ_{Test} sehr variabel, was daran liegt, dass die Daten im Validierungsdatensatz sehr unterschiedlich sein können. Ein weiterer Nachteil dieses Vorgehens besteht darin, dass statistische Methoden dazu tendieren schlechtere Ergebnisse zu liefern, umso geringer die Datenmenge ist, die zur Modellierung benutzt wird. Dies führt auch dazu, dass dieses Vorgehen den Testfehler tendenziell überschätzt [vgl. [Witten et al., 2013] S.178].
Auch bei dem *Leave-One-Out Ansatz* wird der verfügbare Datensatz in zwei Teile aufgeteilt, allerdings nicht in zwei gleich große, sondern in einen Trainigsdatensatz mit $(n-1)$ Beobachtungen und einen Validierungsdatensatz mit einer Beobachtung. Der Trainingsdatensatz dient zur Modellierung und auf Basis der einen Beobachtung, die nicht zur Modellierung verwendet wurde, wird θ_{Test} berechnet. θ_{Test} ist zwar annähernd unverzerrt, dennoch eine eher schlechte Schätzung für den wahren Testfehler, da die Schätzung auf einer einzelnen Beobachtung basiert und dementsprechend eine hohe Varianz aufweist. Um der hohen Varianz entgegenzusteuern, wird dieses Prozedere nun n-mal wiederholt, sodass jede Beobachtung einmal als Validierungsbeobachtung dient. Folglich erhält man n-mal θ_{Test} (θ_{Test_1}, θ_{Test_2}, ..., θ_{Test_n}). Die endgültige Schätzung für den Testfehler berechnet sich dann über

$$\widehat{\theta_{Test}} = \frac{1}{n}\sum_{i=1}^{n}\theta_{Test_i} \quad . \tag{2}$$

Bezüglich des Validierungsdatensatz Ansatzes hat dieses Vorgehen einige Vorteile. Da für die Modellierung $(n-1)$ Datenpunkte verwendet werden, also fast genauso viele wie der gesamte Datensatz enthält, leidet die Modellierung nicht so sehr an mangelnden Daten, was dazu führt, dass die Schätzung für den Testfehler weniger verzerrt ist und tendenziell nicht so stark überschätzt wird [vgl. [Witten et al., 2013] S. 179]. Ein weiterer Vorteil besteht darin, dass die Ergebnisse bei Wiederholen des Vorgehens gleich bleiben und nicht von der Zufälligkeit abhängen, wie der Datensatz aufgeteilt wird.
Bei dem *k-fold Ansatz* wird der vorhandene Datensatz zufällig in k gleich große Teildatensätze aufgeteilt. Hierbei dient einer der Teildatensätze als Testdatensatz und die restlichen $(k-1)$ Teildatensätze als Trainingsdaten. Daraufhin wird anhand der Traingsdatensätze das Modell erstellt und anhand des Testdatensatzes θ_{Test} berechnet. Dieses Vorgehen wird nun k-mal wiederholt, sodass jeder Teildatensatz einmal als Testdatensatz fungiert. Dies führt zu k Schätzungen von θ_{Test} $(\theta_{Test_1}, \theta_{Test_2}, ..., \theta_{Test_k})$. Die endgültige Schätzung für θ_{Test} berechnet sich über den Mittelwert der Schätzungen, also über

$$\widehat{\theta_{Test}} = \frac{1}{k} \sum_{i=1}^{k} \theta_{Test_i} \tag{3}$$

[Witten et al., 2013] [S. 181]. Dieser Ansatz hat bezüglich des Leave-One-Out Ansatzes Vorteile bei der Rechendauer, wenn $k < n$ gewählt wird, da nur k und nicht n Modelle erstellt werden müssen. Doch auch bezüglich der Genauigkeit der Schätzung weist dieses Verfahren Vorteile auf. Die Schätzung für den Testfehler bei dem Leave-One-Out Ansatz ist zwar weniger verzerrt, da für jede Modellierung $(n-1)$ Beobachtungen verwendet werden und nicht nur $(k-1)n/k$ wie bei dem k-fold Ansatz. Will man allerdings die Genauigkeit von Schätzungen bewerten, spielt neben der Verzerrung auch noch die Varianz eine entscheidende Rolle. Bei dem Leave-One-Out Ansatz werden anhand von n Modellen n Testfehler geschätzt und anhand dieser der Durchschnitt bestimmt, welcher dann dem endgültig geschätzten Testfehler entspricht. Da die Modelle aber alle auf fast der gleichen Datenbasis erstellt werden, sind die n geschätzten Testfehler entsprechend stark korreliert. Bei dem k-fold Verfahren werden die Ergebnisse von k erstellten Modellen gemittelt, wobei die Modelle hierbei weniger stark korreliert sind, da die Überschneidung der k verschiedenen Traingsdatensätze nicht so groß ist wie bei dem Leave-One-Out Ansatz. Und da der Mittelwert von stärker korrelierten Werten eine höhere Varianz hat als der Mittelwert von weniger stark korrelierten Werten, tendiert die Schätzung für den Testfehler bei dem Leave-One-Out Ansatz dazu höher zu sein als bei dem k-fold Ansatz [vgl. [Witten et al., 2013] S.183ff]. Zusammenfassend lässt sich sagen, dass der Leave-One-Out Ansatz zwar die

geringste Verzerrung hat, im Vergleich zu dem k-fold Ansatz allerdings eine höhere Varianz besitzt, solange $k < n$ gilt. Es gilt abzuwägen zwischen Verzerrung und Varianz, was bei dem k-fold Ansatz von k abhängt. Empirische Forschungen haben gezeigt, dass der k-fold Ansatz mit $k \in [5; 10]$ die besten Ergebnisse liefert, da die Schätzung für den Testfehler weder unter einer extrem hohen Varianz, noch unter einer extremen Verzerrung leidet [vgl. [Witten et al., 2013] S. 184]. Da der k-fold Ansatz sowohl Vorteile bei der Rechendauer als auch bei der Genauigkeit der Schätzung vorweisen kann, wird dieser auch in späteren Kapiteln dieser Arbeit zur Schätzung verwendet werden.

2.3 Gütemaße

Die Resampling Methoden des vorherigen Kapitels wurden alle exemplarisch anhand der Fehlerrate θ_{Test} erläutert. Dabei funktionieren diese Resampling Verfahren bei anderen Gütemaßen komplett analog. Um die Güte der Modelle zu bewerten, wird mit insgesamt vier verschiedenen Gütemaßen gearbeitet. Eines ist dabei die *Accuracy*. Diese gibt an, welcher Anteil der Testdaten richtig klassifiziert wurde. Dabei liegt dieser Wert im Intervall 0 bis 1, wobei 0 der schlechteste Wert ist, und dem Fall entspricht, dass keine Beobachtung der Testdaten richtig klassifiziert wurde. Liegt die Accuracy bei 1 bedeutet dies, dass alle Testdaten richtig klassifiziert wurden und ist entsprechend der bestmögliche Wert. Die Accuracy lässt sich dabei über $1 - \theta_{Test}$ berechnen. Bezüglich der Berechnung von θ_{Test} siehe Formel (1). Für die folgenden Maße wird die Sicherheit p_{ij} benötigt, welche angibt, wie sicher das Modell die Beobachtung i der Klasse j zuordnet; je höher dabei p_{ij}, umso sicherer ist die prognostizierte Klassenzugehörigkeit der Beobachtung i. Der *logarithmic loss* ist über $-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} y_{ij} log(p_{ij})$ definiert, wobei n für die Anzahl an Beobachtungen steht und k für die Anzahl der Klassen der Responsevariable. y_{ij} ist ein binärer Indikator, der 1 ist, wenn j die wahre Klasse der Beobachtung i ist und sonst 0. Im optimalen Fall werden alle Beobachtungen mit 100% Sicherheit in deren wahren Klassen zugeordnet, sodass der logarithmic loss den Wert 0 annimmt. Im schlechtesten Fall ordnet das Modell alle Beobachtungen mit einer Sicherheit von 0% deren wahrer Klasse j zu, sodass für jedes i der Ausdruck $y_{ij} log(p_{ij})$ extrem klein wird und damit der Gesamtausdruck extrem groß. Je höher also der Wert des logarithmic loss ist, umso schlechter die Güte des Modells. Ein ähnliches Maß stellt der *Multiclass Brier* dar, der sich über $\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}(y_{ij} - p_{ij})^2$ berechnet. y_{ij}, n und k sind dabei genauso definiert wie bei dem Logarithmic loss. Die Güte eines Modells ist am besten, wenn der Multiclass Brier bei 0 liegt, da alle Beobachtungen mit 100% Sicherheit in die richtige Klasse zugeordnet wurden. Der schlechteste Wert

für den Multiclass Brier liegt bei 1 und beschreibt den Fall, dass das Modell alle Beobachtungen mit 0% Sicherheit in ihre wahre Klasse einteilt. Ein weiteres Maß ist die *Multiclass AUC*. Diese entspricht dem AUC ($\hat{=}$ Area Under the ROC Curve), für den Fall, dass der Responsewert mehr als zwei Ausprägungen ($k > 2$) hat und behandelt dabei k-dimensionale Responsewerte als k zweidimensionale Responsewerte. So wird zwischen jedem Responsewert j und den restlichen $(k-1)$ Responsewerten ($rest_j$) unter Berücksichtigung der Priori-Wahrscheinlichkeit ($p(j)$) der AUC berechnet. Der AUC berechnet sich also über $\sum_{j=1}^{k} p(j)AUC(j, rest_j)$. Im optimalen Fall liegt der Wert für den Multiclass AUC bei 1 und im schlechtesten Fall bei 0,5 [vgl. [Ferri et al., 2009] S. 30]. Zur vereinfachten Darstellung wird auch mit Rängen gearbeitet werden. Ein Rang steht für den Platz, den ein Modell belegt, wenn man diese anhand eines Gütemaßes ordnet. Der beste zu erreichende Rang ist der erste, welcher aussagt, dass das Modell den besten Wert bei dem ausgewählten Gütemaß besitzt. Für den Fall, dass zwei Modelle den gleichen Wert für ein Gütemaß besitzen, entscheidet der Zufall, welches Modell welchen Platz bekommt.

2.4 Methodik zur Schätzung der Gütemaße

Die Genauigkeit eines Modelles wird anhand der Gütemaße, die im vorherigen Kapitel vorgestellt wurden, gemessen. Dabei werden die Gütemaße für jedes Modell anhand des k-fold Verfahrens mit $k = 5$, geschätzt (Theorie dazu: Kapitel 2.2). Da die Varianz bei dem k-fold Verfahren recht hoch sein kann, wird dieses Verfahren zehnmal wiederholt, um so die Varianz der Schätzung zu reduzieren. Die entgültige Schätzung für die Gütemaße ist also der gemittelte Wert der zehn Ergebnisse des k-fold Verfahrens mit $k = 5$.

3 Methode der Support Vector Machines

In diesem Abschnitt der Arbeit werden Support Vector Machines sowie die Grundlagen, auf die diese aufbauen, vorgestellt. Support Vector Machines sind statistische Lernmethoden, die sowohl für Regressions- als auch Klassifikationsprobleme angewandt werden können. Diese Arbeit wird sich auf die Anwendung bei Klassifikationsproblemen konzentrieren. Vorgestellt wurde diese Methode 1992 von Boser, Guyon und Vapnik und erfreut sich seitdem großer Beliebtheit, da diese sehr genau ist, auch mit hoch dimensionalen Daten umgehen kann sowie sehr flexibel bei der Modellierung verschiedener Datenquellen ist [vgl. [Ben-Hur and Weston, 2010] S. 223].

3.1 Hyperebenen

Hyperebenen sind flache Unterräume von p-dimensionalen Räumen und haben die Dimension (p-1). Hyperebenen haben in einem p-dimensionalen Raum allgemein die Form:

$$f(x) = \beta_0 + \beta_1 x_1 + ... + \beta_{p-1} x_{p-1} = 0 \quad \text{bzw.} \tag{4}$$

$$f(x) = \beta_0 + x_i^T \beta = 0 \qquad \text{für} \quad i = 1, ..., (p-1) \tag{5}$$

[vgl. [Witten et al., 2013] S. 338]. Am einfachsten vorstellen lässt sich eine Hyperebene in einem zweidimensionalem Raum, da sie dort selber eindimensional ist und eine Gerade darstellt. Sollte ein Vektor $X = (x_1, ..., x_{p-1})^T$ die Gleichung (5) erfüllen, so liegt er auf der Hyperebene, falls jedoch $\beta_0 + x_i^T \beta > 0$ bzw. $\beta_0 + x_i^T \beta < 0$ gilt, so liegt er er über bzw. unter dieser. Man kann sich eine Hyperebene also als Ebene vorstellen, die einen p-dimensionalen Raum in zwei Teile trennt. Diese Trenn-Eigenschaft lässt sich nun zum Klassifizieren nutzen, vorausgesetzt, dass die Daten linear trennbar sind und der Responsewert der Daten binär ist. Der Responsewert, also der Wert, der durch das Modell prognostiziert werden soll, wird meist als y bezeichnet und als Faktor, also als diskrete Variable mit endlichen Ausprägungen, kodiert. So könnte beispielsweise männlich / weiblich mit 1 / -1 kodiert werden. Liegt ein solcher Datensatz vor, lässt sich eine Hyperebene verwenden, die die Einflussvariablen nutzt, um die Daten bezüglich ihres (binären) Responsewertes zu trennen. Dabei erfüllt die sogenannte *trennende Hyperebene* folgende Eigenschaften:

$$\beta_0 + x_i^T \beta > 0, \quad \text{für} \quad y_i = 1 \tag{6}$$

$$\beta_0 + x_i^T \beta < 0, \quad \text{für} \quad y_i = -1 \tag{7}$$

[vgl. [Witten et al., 2013] S. 340]. Datenpunkte, mit $y = 1$ liegen also über der Hyperebene und Datenpunkte mit $y = -1$ darunter, sodass $y_i(\beta_0 + \beta_1 X_{i1} + ... + \beta_p X_{ip}) > 0$ für alle $i = 1, ..., n$ gilt. Dadurch wird garantiert, dass jede Beobachtung auf der richtigen Seite der Hyperebene liegt und der Abstand jeder Beobachtung zu der trennenden Hyperebene größer 0 ist.
Zur Veranschaulichung soll nun ein Beispiel dienen. Sei $\mathbf{X}$ ein Datensatz mit $(p+1)$ Variablen, wobei es p Einflussvariablen gibt und eine Variable den Responsewert y darstellt, wobei dieser binär ist, sodass $y_1, ..., y_n \in \{1, -1\}$ gilt. Sei zudem eine neue Bobachtung x^* vorhanden, die anhand der p Einflussvariablen in eine der zwei Response-Klassen eingeteilt werden soll. Grafisch ist der Datensatz $\mathbf{X}$ samt dessen Klassenzugehörigkeiten sowie der neuen Beobachtung x^* in Abbildung (3) dargestellt. Die drei Geraden in der Abbildung stellen drei von unendlich vielen möglichen, trennenden Hyperebenen

für diese Daten dar. Die neue Beobachtung x^* lässt sich nun mithilfe der trennenden Hyperebenen in eine der zwei Klassen einteilen. Dies geschieht anhand der Hyperebene $f(x) = \beta_0 + x_i^T\beta$. Gilt $f(x^*) < 0$, so wird die neue Bobachtung x^* der Klasse $y = -1$ zugeordnet, falls $f(x^*) > 0$ gilt, entsprechend zu $y = 1$. In dem Beispiel würde der Punkt x^* zu $y = 1$ zugeordnet werden, da er über den Hyperebenen liegt und entsprechend bei allen eingezeichneten trennenden Hyperebenen $f(x^*) > 0$ gilt. Je weiter $f(x^*)$ von 0 entfernt ist, also je weiter entfernt die neue Beobachtung von der trennenden Hyperebene liegt, umso sicherer kann man sich über die Klassenzuordnung der neuen Beobachtung x^* sein.

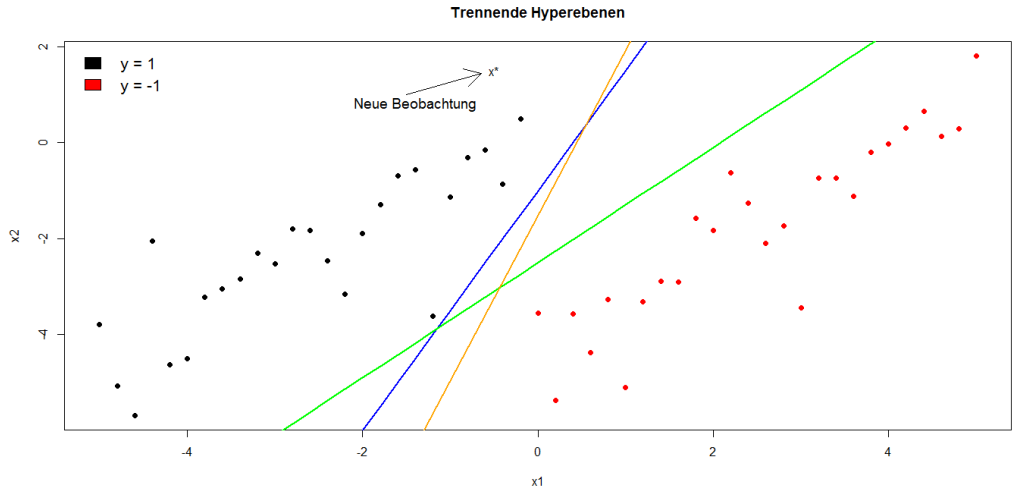


Abbildung 3: Simulierte Datenpunkte und drei verschiedene trennende Hyperebenen sowie eine neue Beobachtung x^*

3.2 Maximal Margin Klassifikator

Wie das Beispiel aus vorherigem Kapitel gezeigt hat, existieren mehr als nur eine trennende Hyperebene, wenn die Daten perfekt linear trennbar sind. Die Frage, die sich nun stellt ist, welche der möglichen trennenden Hyperebenen gewählt werden soll. Da die Zuordnung einer Beobachtung in eine Klasse umso sicherer ist, je weiter ein Datenpunkt von der trennenden Hyperebene entfernt ist, ist es entsprechend sinnvoll eine trennende Hyperebene zu wählen, deren Abstand zu den Trainigsdaten sehr groß ist. Der Begriff *Margin* bezeichnet hierbei den Mindestabstand zwischen Hyperebene und Trainingsdaten, sodass alle Trainingsdaten mindestens so weit von der Hyperebene entfernt liegen, wie die Margin der trennenden Hyperebene breit

ist. Diese Eigenschaft wird bei der *Maximal Margin Hyperebene* genutzt, sodass diese genau diejenige trennende Hyperebene darstellt, deren Margin am breitesten ist. Dies entspricht derjenigen Hyperebene, die die Distanz der am nähsten zueinander liegenden Punkte unterschiedlicher Klassenzugehörigkeit maximiert. Hierbei ist die Idee, dass eine breite Margin bei den Trainingsdaten zu einer guten Trennung bei den Testdaten führt [vgl. [Hastie et al., 2016] S.132ff]. Bei den drei trennenden Hyperebenen in Abbildung (3) ist die Margin der blauen trennenden Hyperebene am geringsten und die der orangen am größten, sodass die orange trennende Hyperebene als Maximal Margin Hyperebene gewählt würde. Diese ist in Abbildung (4) dargestellt. Die gestrichelten Linien, die symmetrisch um die Maximal Margin Hyperebene liegen, stellen die Margin dar. Insgesamt liegen drei Punkte auf der Margin des Klassifikators. Diese drei Punkte haben alle den geringsten Abstand zu der Maximal Margin Hyperebene und werden als *Support Vektoren* bezeichnet.

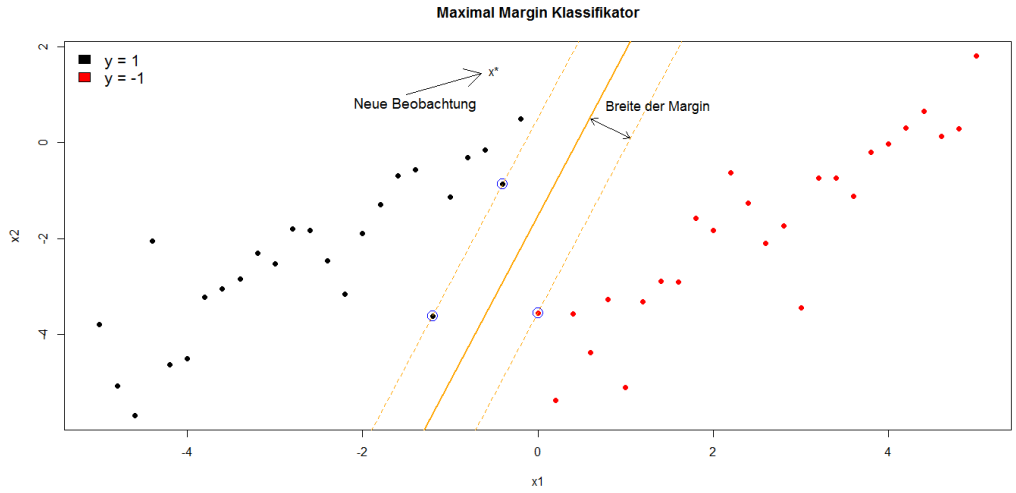


Abbildung 4: Simulierte Datenpunkte zur Darstellung eines Maximal Margin Klassifikators, in gestrichelten Linien dessen Margin sowie dessen eingekreiste Support Vektoren. Zudem eine neue Beobachtung x^*.

Der Name Support Vektor rührt daher, dass diese Datenpunkte Vektoren in einem p-dimensionalen Raum darstellen, von der die Maximal Margin Hyperebene direkt abhängt. Ändert sich die Lage der Support Vektoren, ändert sich auch die Lage der Maximal Margin Hyperebene. Ändert sich allerdings die Lage von anderen Datenpunkten, verändert sich die Lage der Maximal Margin Hyperebene nicht, vorausgesetzt der Abstand zwischen dem Datenpunkt und der Hyperebene bleibt größer als die Margin breit ist. Das Vor-

gehen, um neue Beobachtungen anhand der Maximal Margin Hyperplane zu klassifizieren, erfolgt analog zu den trennenden Hyperebenen. Entsprechend würde auch die neue Beobachtung x^* zu Klasse 1 zugeordnet werden, da $f(x^*) > 0$ gilt, wobei f(x) nun die Maximal Margin Hyperebene repräsentiert. Wird nun die Maximal Margin Hyperebene zur Klassifizierung von neuen Beobachtungen genutzt, spricht man von dem *Maximal Margin Klassifikator* (MMK). Dieser Maximal Margin Klassifikator lässt sich anhand der folgenden Optimierungsprobleme konstruieren:

$$\max_{\beta_0,\beta,||\beta||=1} M \tag{8}$$

unter der Nebenbedingung

$$y_i(\beta_0 + x_i^T\beta) \geq M \quad \forall i = 1, ..., n \tag{9}$$

[Witten et al., 2013] [S. 343]. Die Bedingung aus Formel (9) garantiert, dass alle Beobachtungen auf der richtigen Seite der Hyperebene liegen und deren Abstand zu der Hyperebene größer gleich M ist. Hierbei repräsentiert M die Breite der Margin der Hyperebene und ist größer 0. Nach (8) werden β_0 und β, die die Hyperebene festlegen, so gewählt, dass M maximal wird, was genau der Definition einer Maximal Margin Hyperebene entspricht [vgl. [Hastie et al., 2016] S.132].
Das Problem hierbei ist allerdings, dass eine trennende Hyperebene existieren muss, um einen Maximal Margin Klassifikator zu konstruieren und es durchaus vorkommen kann, dass die Daten nicht perfekt linear trennbar sind und es deshalb keine trennende Hyperebene gibt. Folglich existiert in diesen Fällen auch kein Maximal Margin Klassifikator. In Abbildung (5) ist ein Beispiel dargestellt, in dem sich die Daten nicht perfekt trennen lassen. Die abgebildeten Daten sind nahezu identisch mit den Daten aus Abbildung (4), nur ein einziger Datenpunkt wurde ergänzt. Dieser einzelne Punkt ist Grund dafür, dass keine trennende Hyperebene existiert, da sich die Daten nun nicht mehr linear durch eine Hyperebene trennen lassen.

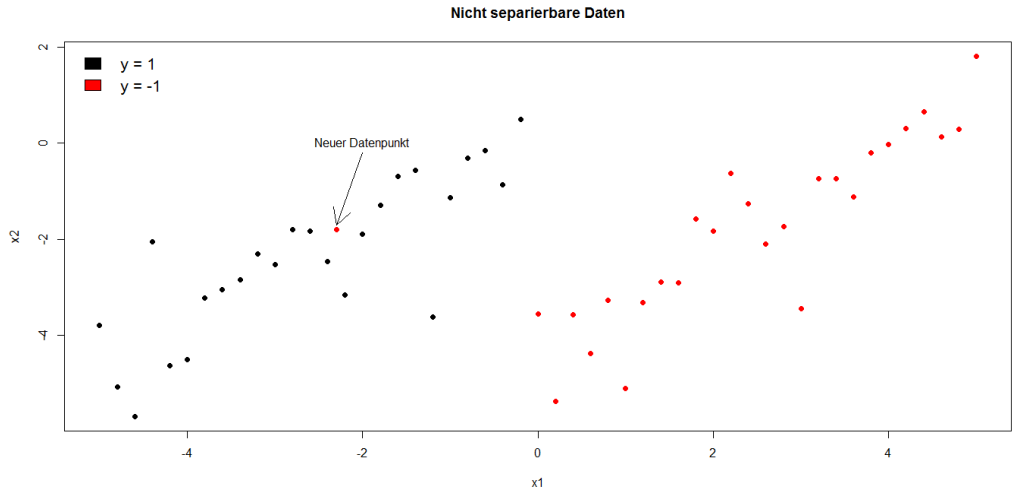


Abbildung 5: Verteilung eines binären Merkmals. In diesem Fall existiert keine trennende Hyperebene, da sich die Daten nicht perfekt linear trennen lassen.

Da der Maximal Margin Klassifikator die Daten so perfekt trennt, dass jede Beobachtung auf der richtigen Seite der Hyperebene liegt, führt dies dazu, dass die Hyperebene nicht robust, beziehungsweise sehr sensitiv gegenüber einzelnen Punkte sein kann. Dieses Fehlen der Robustheit gegenüber einzelner Punkte wird exemplarisch in Abbildung (6) dargestellt. Auf der linken Seite der Abbildung (6) ist die Maximal Margin Hyperebene aus Abbildung (4) dargestellt. Rechts ist noch einmal eine Maximal Margin Hyperebene zu sehen. Allerdings wurde hierbei, im Vergleich zu den Daten auf der linken Seite, ein einziger Datenpunkt ergänzt. Dieser einzelne Punkt sorgt sowohl dafür, dass sich die Lage der Maximal Margin Hyperebene drastisch ändert, als auch dazu, dass die Margin wesentlich schmaler wird. Die Maximal Margin Hyperebene kann gegenüber einzelnen Punkten also extrem sensitiv sein, was dazu führen kann, dass sich die Lage extrem verändert. Es ist ersichtlich, dass die Methode des Maximum Margin Klassifikators sehr empfindlich gegenüber einzelnen Datenpunkten ist und dies die Vermutung nahe legt, dass der Maximum Margin Klassifikator anfällig für eine Überanpassung (Overfitting) an die Trainingsdaten ist.

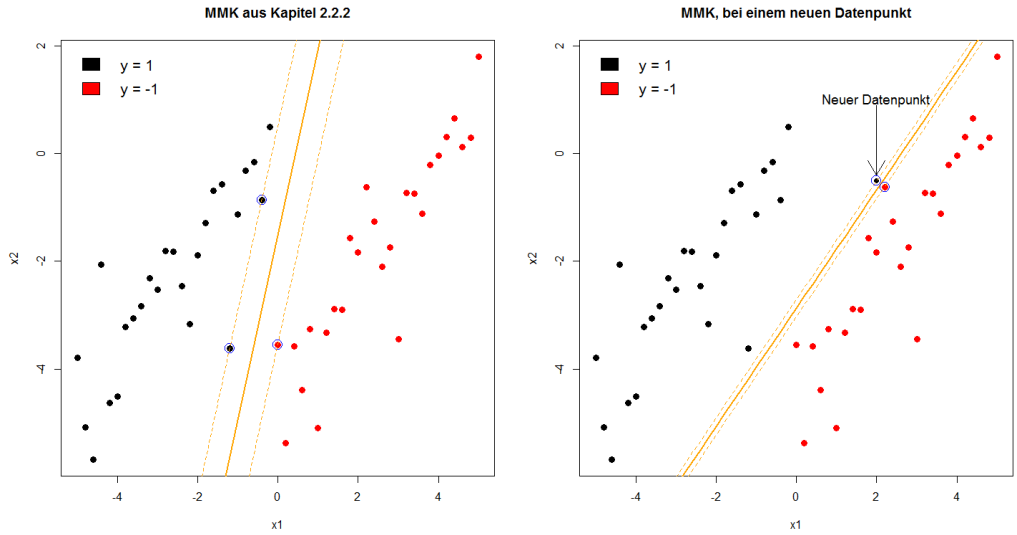


Abbildung 6: Links: Verteilung eines binären Merkmals sowie der dazugehörigen Maximal Margin Hyperebene. Rechts: Verteilung eines binären Merkmals, wobei ein neuer Datenpunkt zu der Klasse y = 1 hinzugefügt wurde. Der neue Datenpunkt führt zu einer drastischen Veränderung der Lage der Maximal Margin Hyperebene.

3.3 Support Vektor Klassifikator

Wie im vorherigen Kapitel festgestellt wurde, besitzt der Maximal Margin Klassifikator einige entscheidende Nachteile, sodass die Qualität des Klassifikators stark davon abhängt, mit welchen Daten er erstellt wird. Eine Lösung dieser Probleme stellt der *Support Vektor Klassifikator*, kurz SVK, dar. Dieser ist robuster gegenüber einzelnen Beobachtungen, da er nicht versucht die Daten perfekt zu trennen, sondern auch eine gewisse Anzahl an Fehlern akzeptiert. Das ist auch der Grund, weshalb der SVK auch mit Daten umgehen kann, die nicht perfekt trennbar sind. Es werden bei dem SVK also eine gewisse Anzahl an Fehlern in Kauf genommen, wenn sich dafür auf der anderen Seite die Klassifizierung der meisten Datenpunkte verbessert [vgl. [Witten et al., 2013] S.345]. Dabei wird bei dieser Methode zwischen zwei Arten von Fehlern unterschieden: Auf der einen Seite ist es ein Fehler, wenn ein Datenpunkt auf der falschen Seite der Hyperebene liegt. Und auf der anderen Seite ist es auch ein Fehler, wenn der Datenpunkt zwar auf der richtigen Seite, aber näher an der Hyperebene liegt als die Margin der Hyperebene, der Punkt also zwischen Hyperebene und Margin liegt. Die Klassifizierung neuer Beobachtungen mit dem SVK erfolgt analog zu dem Maximal Margin

Klassifikator. Eine neue Beobachtung wird also abhänging davon, auf welcher Seite der Hyperebene sie liegt, klassifiziert.
Die Konstruktion einer Maximal Margin Hyperebene beruht darauf, dass immer noch die Breite der Margin maximiert wird, aber einer gewissen Anzahl an Trainingsdaten erlaubt wird, auf der falschen Seite der Hyperebene beziehungsweise Margin zu liegen, sodass das Optimierungsproblem folgende Form annimmt:

$$\max_{\beta_0,\beta,||\beta||=1} M \tag{10}$$

unter den Nebenbedingungen

$$y_i(\beta_0 + x_i^T\beta) \geq M(1-\epsilon_i) \quad \forall i = 1, ..., n \tag{11}$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C \quad \forall i = 1, ..., n \tag{12}$$

[vgl. [Witten et al., 2013] S. 346]. Dabei werden die ϵ_i's als Schlupfvariablen bezeichnet, sind größer gleich 0 und geben Information darüber, wo die i-te Beobachtung liegt. So liegt die i-te Beobachtung sowohl auf der richtigen Seite der Hyperebene als auch auf der richtigen Seite der Margin, wenn $\epsilon_i = 0$ gilt. Für $\epsilon_i \in (0; 1]$ liegt die i-te Beobachtung auf der falschen Seite der Margin, aber noch auf der richtigen Seite der Hyperebene und für $\epsilon_i > 1$ liegt die i-te Beobachtung auf der falschen Seite der Hyperebene. Umso höher ϵ_i, umso schwerwiegender ist also der Fehler. Der Parameter C aus (12) stellt einen nicht negativen Hyperparameter dar, der die Summe der ϵ_i's , und damit die Summe der Fehler und deren Schwere, begrenzt. Für den Fall, dass die Daten perfekt trennbar sind, lässt sich $C = 0$ setzen, sodass kein einziger Fehler zugelassen wird und entsprechend $\epsilon_1 = ... = \epsilon_n = 0$ gilt. In diesem Fall entspricht der Support Vektor Klassifikator dem Maximal Margin Klassifikator. Der Hyperparameter C hat einen kontrollierenden Einfluss auf das Abwägen von Verzerrung und Varianz. Je größer C ist, umso mehr Fehler werden akzeptiert und dadurch wird die Margin der Hyperebene breiter. Dementsprechend muss sich das Modell nicht so stark an einzelne Datenpunkte anpassen, sodass die Varianz zwar geringer, die Gefahr einer Verzerrung dafür größer ist. Folglich wird bei kleinerem C die Methode sensibler gegenüber Fehlern, was dazu führt, dass die Margin schmaler wird und dass sich das Modell stärker an einzelne Datenpunkte anpassen muss. Folglich ist die Verzerrung bei kleinem C eher gering, dafür die Varianz deutlich höher [vgl. [Witten et al., 2013] S. 347]. Genauso wie bei dem Maximal Margin Klassifikator haben auch nur die Support Vektoren Einfluss auf die Lage der

Hyperebene, allerdings sind die Support Vektoren bei dem Support Vektor Klassifikator nicht nur diejenigen Punkte, die auf der Margin liegen, sondern auch diejenigen, die auf der falschen Seite der Margin, beziehungsweise auf der falschen Seite der Hyperebene liegen. Dementsprechend führt ein breiter Rand, also ein großes C, dazu, dass mehr Support Vektoren existieren.
In Abbildung (7) sind zwei Support Vektor Klassifikatoren mit unterschiedlichem Hyperparameter C dargestellt. An diesen lässt sich der Einfluss des Hyperparameters C sehr gut erkennen. Die Daten, die zur Veranschaulichung verwendet werden, haben keine klare lineare Trennung. Modell 1 in Abbildung (7) stellt ein Modell mit sehr hohem Hyperparameter C dar, sodass die Margin der Hyperebene sehr breit ist und entsprechend viele Support Vektoren existieren. Bei dem zweiten Modell mit dem geringeren Hyperparameter C ist der Rand der Hyperebene wesentlich schmaler als in Modell 1, sodass auch entsprechend weniger Support Vektoren existieren. Neben der Anzahl an Support Vektoren und der Breite des Randes unterscheiden sich die zwei Hyperebenen auch in deren Lage, was auch auf den Hyperparameter C zurückzuführen ist.

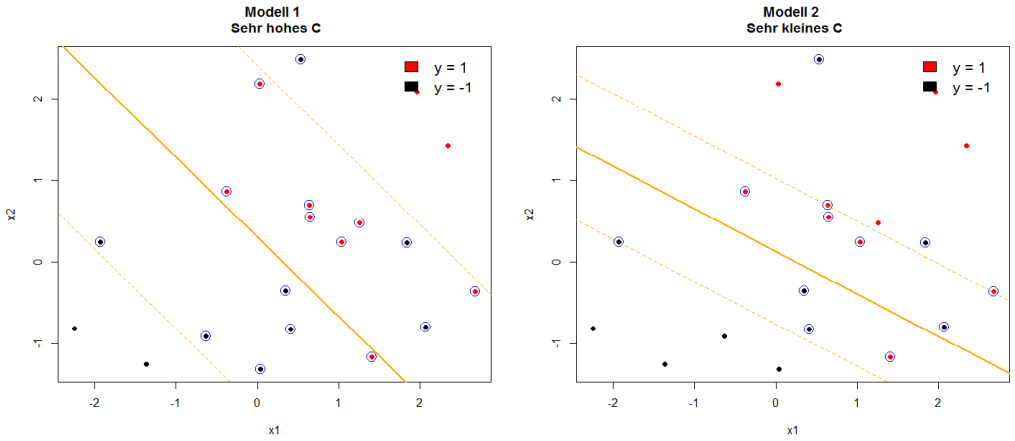


Abbildung 7: Links: SVK mit einem hohen Wert für Hyperparameter C und folglich mehr Support Vektoren als in Modell 2. Rechts: SVK mit geringem Wert für Hyperparameter C und folglich weniger Support Vektoren als in Modell 1.

3.4 Support Vector Machines

Alle Verfahren, die seit den Kapiteln 3.1 vorgestellt wurden, gehen von einer linearen bzw. annähernd linearen Trennbarkeit der Daten aus. Es kann aber durchaus vorkommen mit Daten konfrontiert zu sein, die sich nicht, beziehungsweise nur sehr schlecht linear trennen lassen. Blicken wir auf das Beispiel aus Kapitel 2.1 zurück und betrachten den Scatterplot der Daten aus Abbildung (1), beziehungsweise den Scatterplot links oben in Abbildung (8), so sollte offensichtlich sein, dass diese Daten nicht adäquat durch einen linearen Klassifikator getrennt werden können. Dementsprechend würde man bei der Anwendung eines linearen Klassifikators auf diese Daten einen sehr hohen Testfehler θ_{Test} erwarten. Eine Möglichkeit, um das Problem dieser nicht linearen Grenzen zwischen den Daten anzugehen, ist die Vergrößerung des Merkmalraumes durch beispielsweise höher geordnete Polynome oder Interaktionsterme. Dies führt dazu, dass sich der Merkmalsraum, anhand dessen die Daten klassifiziert werden sollen, vergrößert und in diesem erweiterten Merkmalsraum eine lineare Trennung der Daten möglich ist. Diese lineare Trennung in dem erweiterten Merkmalsraum führt zu einer besseren Trennung der Trainingsdaten und entspricht einer nicht linearen Trennung im ursprünglichen Merkmalsraum [vgl. [Hastie et al., 2016] S.423]. Die Möglichkeiten den Merkmalsraum zu erweitern, sind dabei nahezu unbegrenzt, allerdings kann es hierbei passieren, dass der Merkmalsraum extrem groß wird und damit die Berechnungen extrem zeitaufwändig werden. Eine effiziente Möglichkeit zur Erweiterung des Merkmalsraumes stellt die Methode der *Support Vector Machines*, kurz SVM, dar. Die Theorie, die hinter diesen steckt, soll in dem folgenden Absatz erläutert und anschließend anhand von Beispielen illustriert werden.

3.4.1 Theorie von Support Vector Machines

Die Support Vector Machines stellen eine Erweiterung des Support Vektor Klassifikators dar, wobei diese den Merkmalsraum durch sogenannte *Kernel* erweitern. Dies hat den Effekt, dass auch nicht lineare Grenzen in den Daten adäquat modelliert werden können. Ein entscheidender Punkt, um zu verstehen, wie Kernel funktionieren, ist der, dass die Lösung für das Optimierungsproblem für Support Vektor Klassifikatoren nur anhand des inneren Produktes der Beobachtungen gelöst werden kann. Dabei lässt sich dieses Optimierungsproblem aus (10) - (12) wie folgt umschreiben:
Die Normierung $||\beta|| = 1$ aus (10) lässt sich umgehen, indem man die Nebenbedingung (11) zu $y_i(\beta_0 + x_i^T\beta) \geq M(1 - \epsilon_i)||\beta|| \quad \forall i = 1, ..., n$ umformt.

Dabei ist diese Ungleichung für jedes β_0, β wahr, sodass $||\beta||$ auf $1/M$ gesetzt werden kann [vgl. bis hierhin [Hastie et al., 2016] S.132]. Diese Umformung hat zur Folge, dass sich die Bedingungen aus Formel (10) - (12) äquivalent als

$$\min ||\beta|| \tag{13}$$

unter den Nebenbedingungen

$$y_i(\beta_0 + x_i^T\beta) \geq 1 - \epsilon_i \quad \forall i = 1, ..., n \tag{14}$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C \quad \forall i = 1, ..., n \tag{15}$$

darstellen lassen. Zur Lösung dieses konvexen Optimierungsproblems wird mithilfe von Lagrange Multiplikatoren gearbeitet. Dabei lassen sich (13) - (15) in äquivalenter Form darstellen durch

$$\min_{\beta_0,...,\beta_p} \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n} \epsilon_i \tag{16}$$

unter der Nebenbedingung

$$\epsilon_i \geq 0, \quad y_i(\beta_0 + x_i^T\beta) \geq 1 - \epsilon_i \quad \forall i = 1, ..., n \quad . \tag{17}$$

Diese Umformung ist nötig, da sich dieses Optimierungsproblem rechentechnisch leichter lösen lässt. Durch Minimierung der primalen Langrange Funktion und dem Einsetzen der dadurch berechneten β_i's und α_i's, erhält man die Duale Wolfe Lagrange Funktion der Form

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{n} \alpha_i\alpha_{i'}y_iyi'x_i^Tx_{i'} \quad . \tag{18}$$

Dieses Optimierungsproblem lässt sich unter den Nebenbedingungen $\alpha_i \in [0; C]$ und $\sum_{i=1}^{n} \alpha_i y_i = 0$ mittels Standardtechniken maximieren [vgl. bis hierhin [Hastie et al., 2016] S.419ff].
Dabei lässt sich das Optimierungsproblem auch mit transformierten Einflussvariablen $h(x_i)$ darstellen, so dass die duale Wolfe Lagrange Funktion aus (18) durch die Ersetzung der Einflussvariablen durch die transformierten Einflussvariablen folgende Form annimmt:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{n} \alpha_i\alpha_{i'}y_iyi' < h(x_i), h(x_{i'}) > \tag{19}$$

Dabei lässt sich dieses Optimierungsproblem für geeignet gewählte $h(x)$ effizient lösen. Die Transformationen h(x) müssen dabei nicht selber festgelegt werden, sondern es wird nur eine Kernel Funktion benötigt, die durch das innere Produkt$< h(x_i), h(x_{i'}) >$ dargestellt wird, sodass

$$K(x, x') =< h(x), h(x') > \tag{20}$$

gilt. Durch die Wahl verschiedener Kernel kann der Merkmalsraum, in dem die Daten getrennt werden sollen, effizient erweitert werden. Der einfachste Kernel ist der *lineare Kernel*, bei dem der Kernel genau die Form wie in (20) annimmt. Bei diesem Kernel wird der Merkmalsraum also nicht erweitert. Ein weiterer Kernel stellt der *Polynomiale Kernel* dar, der die Form

$$K(x, x') = (1+ < x, x' >)^d \tag{21}$$

annimmt, wobei $d \in \mathbb{N}^+$ gilt und einen Hyperparameter für diesen Kernel darstellt. Für $d > 1$ führt dieser Kernel zu flexibleren Entscheidungsgrenzen. Hierbei ist die Gefahr einer Überanpassung an die Daten sehr groß, wenn der Hyperparameter d zu hoch gewählt wird, da der Merkmalsraum dadurch sehr stark erweitert wird.
Eine weitere sehr populäre Wahl für einen nicht linearen Kernel, wie auch der polynomiale Kernel einer ist, stellt der *Radiale Kernel* dar. Dieser hat die Form

$$K(x, x') = exp(-\gamma||x - x'||^2) \quad . \tag{22}$$

Hierbei stellt γ einen Hyperparameter für diesen Kernel dar, wobei $\gamma \in \mathbb{R}^+$ gilt [vgl. bis hierhin [Hastie et al., 2016] S.423f]. Dabei hat dieser Kernel ein sehr lokales Verhalten. Dies liegt daran, dass die euklidische Distanz bei weit auseinanderliegenden Punkten sehr groß ist und damit der gesamte Kernel sehr klein wird, sodass eine Trainingsbeobachtung, die weit entfernt von der neuen Beobachtung liegt, kaum einen Einfluss auf die Klassifizierung der neuen Beobachtung hat. Je größer γ gewählt wird, umso stärker wird die Entfernung gewichtet, sodass ein großes γ dazu führt, dass der Klassifikator mit radialem Kernel noch lokaler wirkt. Wird der Support Vektor Klassifikator mit einem nicht linearen Kernel (Polynomialen/ Radialen) kombiniert, so bezeichnet man die dabei entstehenden Klassifikatoren als Support Vector Machines [vgl. [Witten et al., 2013] S. 352f].
Der Einfluss des Hyperparameters d auf den Polynomialen Kernel und des Hyperparameters γ auf den Radialen Kernel werden im Verlauf des nächsten Kapitels genauer erläutert.

3.4.2 Einfluss von Kerneln und deren Hyperparameter

In diesem Abschnitt der Arbeit soll veranschaulicht werden, welchen Einfluss die verschiedenen Kernel sowie deren Hyperparameter haben. Dazu werden simulierte Daten verwendet, deren Verteilung links oben in Abbildung (8) dargestellt ist. Dabei gilt für die in schwarz dargestellten Datenpunkte $X_1, .., X_{75} \sim N(\pm 2.5; 0)$, diejenigen Punkte in rot $X_{76}, ..., X_{100}$ folgen einer Standardnormalverteilung. Die Daten fallen in zwei verschiedene Klassen, sodass die schwarzen Datenpunkte zu der Klasse $y = -1$ gehören und die roten zu der Klasse $y = 1$. Zur Darstellung der Klassifikatoren werden nun andere Grafiken als bisher verwendet werden. Diese Grafiken folgen alle dem gleichen Muster und unterscheiden sich nur bezüglich ihrer Modelle, auf deren Basis sie erstellt werden. Bei dieser Darstellungsform von Klassifikatoren wird der Merkmalsraum in farbige Bereiche aufgeteilt, wobei jede Farbe eine Klasse repräsentiert. Neben dem eigentlichen Plot ist noch eine kleine Grafik, die zeigt, welche Farbe welche Gruppe repräsentiert. In diesem Abschnitt der Arbeit stellt grau die Farbe für die Klasse $y = -1$ dar und oliv die Farbe für die Klasse $y = 1$. Fällt eine neue Beobachtung, von der nur x.1 und x.2 bekannt sind, in den oliven Bereich, so wird diese neue Beobachtung zur Klasse $y = 1$ zugeordnet, fällt sie in den grauen Bereich entsprechend zu $y = -1$. Die roten Datenpunkte stellen Trainingsdaten aus der Klasse $y = 1$ dar und die schwarzen Trainingsbeobachtungen aus der Klasse $y = -1$. Folglich sind schwarze Datenpunkte im oliven Bereich, beziehungsweise rote Datenpunkte im grauen Bereich missklassifizierte Trainingsdaten. Datenpunkte, die als Kreuze und nicht als Kreise dargestellt sind, sind die Support Vektoren des Modells.
Im folgenden soll der Einfluss verschiedener Kernel untersucht werden. Dazu wurden auf Basis der simulierten Daten, die links oben in Abbildung (8) dargestellt sind, drei verschiedene Modelle erstellt, die alle den gleichen Hyperparameter $C = 10$ besitzen und sich nur in deren Kernel unterscheiden. Hierbei gilt es zu beachten, dass die Anzahl der Fehler, die das Modell machen, darf nicht durch C, sondern durch die Form $\frac{1}{C}$ festgesetzt wird. Folglich sind die Modelle mit kleinerem C weniger sensibel gegenüber Fehlern als Modelle mit großem C. Dies entspricht genau dem Gegenteil dessen, was in Kapitel 3.3 Formel (12) vorgestellt wurde, und ist auf programmiertechnische Gründe zurückzuführen.
Bei Betrachtung des Scatterplots der simulierten Daten links oben in Abbildung (8) sollte ersichtlich sein, dass eine lineare Trennung der Daten hier vermutlich keine guten Ergebnisse liefern würde, da eine solche in dem ursprünglichen Merkmalsraum nicht möglich ist. Bei Betrachtung des Klassifikators mit linearem Kernel rechts neben dem Scatterplot bestätigt sich diese

Vermutung. Der Klassifikator ist nicht fähig die Daten zu teilen, sodass der gesamte Merkmalsraum grau ist und entsprechend zu der Klasse $y = -1$ gehört. Das ist in diesem Fall natürlich nicht befriedigend, da er die 25 Trainingsdatenpunkte aus Klasse $y = 1$ komplett missklassifiziert und dieses Modell für Prognosen folglich nicht geeignet zu sein scheint. Das Modell mit dem Polynomialen Kernel ist links unten in Abbildung (8) zu sehen. Mit diesem Kernel ist die Aufteilung des Merkmalsraums in zwei Bereiche möglich und scheint dabei die Struktur der Daten recht gut zu erfassen. Zur Veranschaulichung des radialen Kernels betrachten wir die letzte Grafik rechts unten in Abbildung (8). Auch mit diesem Kernel wird die Struktur der Daten gut erkannt und scheint mit den nicht linearen Grenzen zwischen den Daten keine Probleme zu haben. Es lässt sich also zusammenfassend festhalten, dass der lineare Kernel Datenstrukturen nur linear trennen kann und der Polynomiale Kernel hierbei etwas flexibler ist. Am flexibelsten scheint der Klassifikator mit Radialem Kernel zu sein.

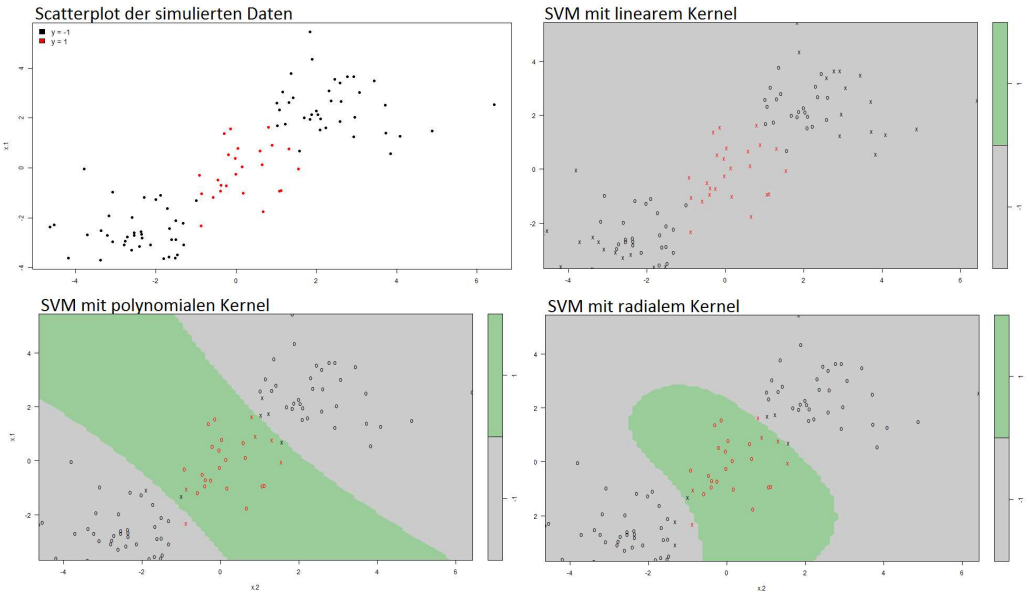


Abbildung 8: Links Oben: Verteilung von simulierten Daten anhand derer die Modelle erstellt wurden. Rechts Oben: Der Support Vektor Klassifikator. Links Unten: Die Support Vector Machine mit Polynomialem Kernel. Rechts Unten: Die Support Vector Machine mit radialem Kernel.

Die Art der Klassifikation hängt bei dem Radialen und dem Polynomialen Kernel allerdings noch von weiteren Parametern ab und ist anders als der lineare Kernel nicht nur von dem Hyperparameter C abhängig. Diesen Einfluss der Kernelspezifischen Hyperparameter soll nun im folgenden anhand des simulierten Datensatzes, der bereits in diesem Abschnitt verwendet wurde, untersucht werden.
Bei dem Polynomialen Kernel gibt es neben dem Hyperparameter C, der die Anzahl der Fehler und deren Schwere begrenzt, noch den weiteren Hyperparameter d, der den Grad des Polynomes festlegt. In Abbildung (9) sind vier verschiedene Polynomiale Klassifikatoren dargestellt. Für die Modelle 1 bis 3 ist der Hyperparameter $C = 10$, sodass sich diese Modelle nur in ihrem Hyperparameter d unterscheiden. Modell 1 mit $d = 2$ und Modell 3 mit $d = 8$ erkennen die Struktur der Daten recht gut, wobei Modell 1 etwas ruhiger verläuft und nicht einen solchen Bauch wie Modell 3 hat. Bei Modell 2 wurde der Hyperparameter d auf 3 gesetzt. Dieses Modell schafft es nicht die Daten zu trennen, sodass der Eindruck entsteht, dass die Daten in dem erweitertem Merkmalsraum mit ungeradem d nicht getrennt werden können. Doch nicht nur der Hyperparameter d hat einen Einfluss darauf, wie die Grenzen des Modelles verlaufen, sondern natürlich auch der Hyperparameter C. Dessen Einfluss wird bei Betrachtung der Modelle 3 und 4 ersichtlich. Beide Modelle unterscheiden sich nur in dem Hyperparameter C, wobei der des Modelles 4 1 Milliarde mal höher liegt als der des Modells 3. Folglich ist bei der Erstellung von Modell 4 nur eine sehr geringe Anzahl an Fehlern erlaubt, sodass sich die Grenzen dieses Modells stark an einzelnen Trainingsdatenpunkten orientieren und entsprechend unruhig verlaufen. Modell 3 hingegen besitzt einen wesentlich geringeren Hyperparameter C, sodass bei der Modellerstellung mehr Fehler akzeptiert werden, was zur Folge hat, dass die Grenzen wesentlich ruhiger verlaufen.

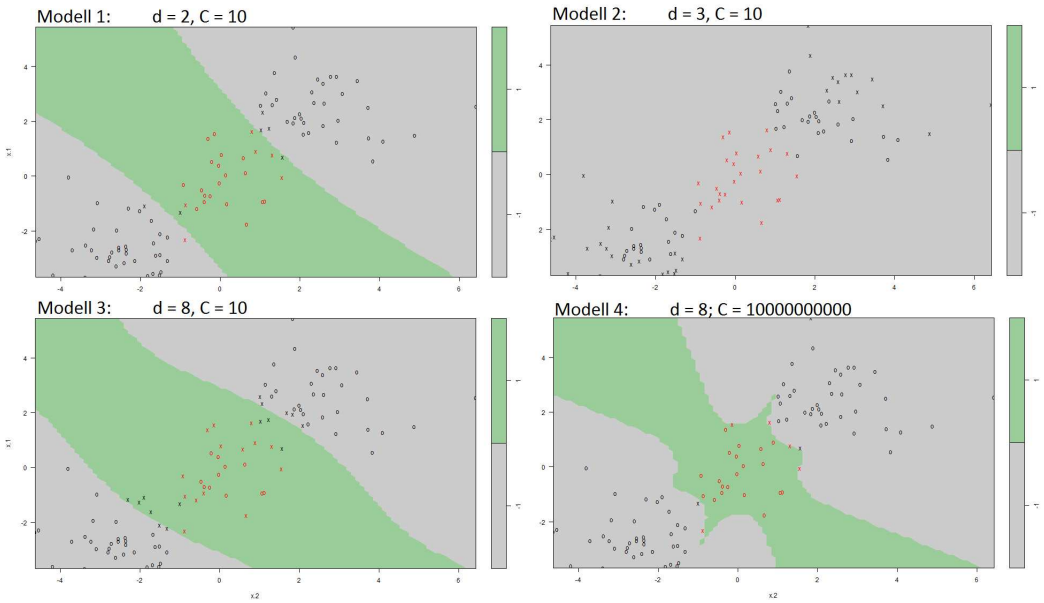


Abbildung 9: Vier verschiedene Modelle mit Polynomialem Kernel, die sich nur bezüglich ihrer Hyperparameter unterscheiden

Bei dem Radialen Kernel gibt es neben dem Hyperparameter C noch den weiteren Hyperparameter γ. In Abbildung (10) sind vier verschiedene Radiale Klassifikatoren dargestellt. Der Hyperparameter C wurde für die Modelle 1 bis 3 auf fünf gesetzt, sodass sich diese Modelle nur in ihrem Hyperparameter γ unterscheiden. Modell 1 hat dabei mit $\gamma = 1$ den niedrigsten Hyperparameter und erkennt die Struktur der Daten recht gut. Modell 3, bei dem γ auf 5 gesetzt wurde, erkennt die Struktur auch ziemlich gut, verhält sich allerdings etwas lokaler als Modell 1. Modell 2 hat mit $\gamma = 100$ den höchsten Wert für γ und verhält sich extrem lokal, sodass der der Klassifikator auschließlich um die Punkte der entsprechenden Klassen läuft und die Grenzen sehr unruhig verlaufen. Modell 3 und 4 unterscheiden sich nur in ihrem Hyperparameter C, wobei C bei Modell 4 200 Milliarden mal höher gesetzt wurde als in Modell 3, was in diesem Fall allerdings nur zu einer marginalen Veränderung führt.

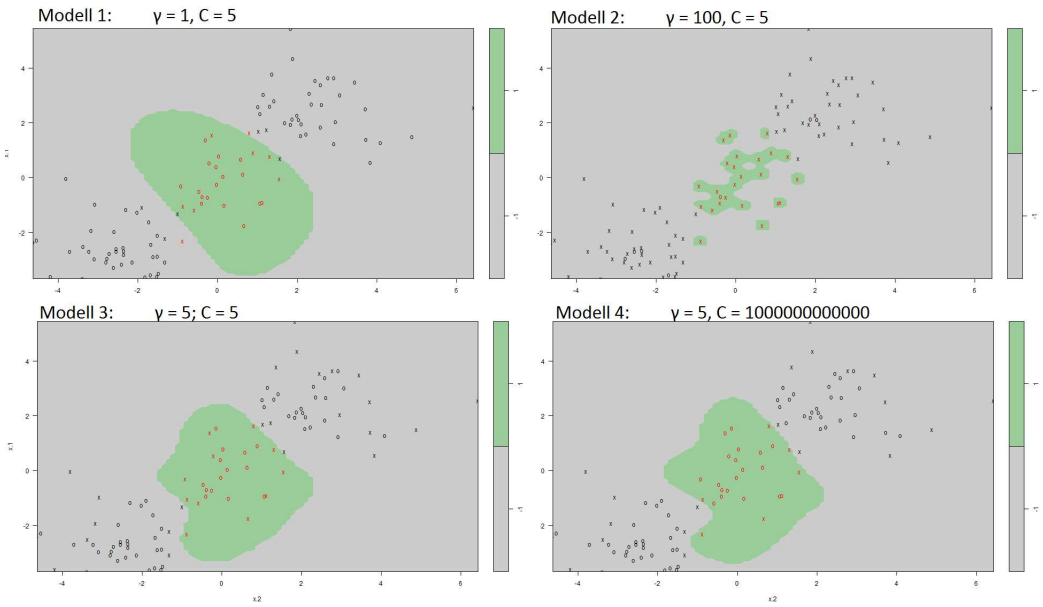


Abbildung 10: Vier verschiedene Modelle mit Radialem Kernel, die sich nur bezüglich ihrer Hyperparameter unterscheiden

3.4.3 Mehrdimensionale Responsewerte

Bisher wurde nur der Fall betrachtet, dass der Response unseres Datensatzes binär ist. Doch auch Support Vector Machines können mit Daten umgehen, deren Response K Kategorien ($K > 2$) hat. Dafür werden $\binom{K}{2}$ Modelle erstellt, von denen jedes Modell zwei Kategorien des Responsewertes vergleicht. Eine neue Beobachtung wird anhand der $\binom{K}{2}$ Modelle klassifiziert und schlussendlich der Klasse zugewiesen, zu der sie in den $\binom{K}{2}$ Klassifikationen am häufigsten zugewiesen wurde [vgl. [Witten et al., 2013] S. 355f]

4 Datensätze

In diesem Teil der Arbeit wird auf die Daten eingegangen, mit denen der Einfluss der Hyperparameter auf Support Vector Machines untersucht wird.

4.1 Auswahl der Datensätze

Die Datensätze, mit denen gearbeitet wird, sind von der Online Plattform *openml.org*, wobei `OpenML` für Open Machine Learning steht. Dabei handelt es sich um ein offenes Wissenschaftsprojekt, das das Teilen von Daten, Code und Machine Learning Experimenten ermöglicht [vgl. [Vanschoren et al.,

2015]]. Die Daten, die OpenML zur Verfügung stellt, enthalten neben den eigentlichen Daten noch Informationen darüber, für welche Art von Machine Learning der Datensatz vorgesehen ist, ob es fehlende Werte gibt, wie viele Einflussgrößen es gibt etc.. Vor den Analysen mussten die Datensätze auf die Methode der Support Vector Machines angepasst werden. Dabei war das erste große Auswahlkriterium, dass die Datensätze in die Klasse der überwachten Klassifizerung fallen, dass also die Responsevariable des Datensatzes bekannt ist und diese eine diskrete Variable mit endlichen Ausprägungen darstellt. Zudem sollten die Daten keine fehlenden Werte enthalten, da Support Vector Machines mit fehlenden Daten nicht umgehen können. Da die Annahmen für die Imputation, dass die fehlenden Werte zufällig fehlen und keinem Muster folgen [missing at random Annahme], bei so vielen Datensätzen schwer überprüft werden hätte können, wurden die Datensätze mit fehlenden Werten aussortiert. Zuletzt wurden doppelte und simulierte Datensätze aussortiert, sodass am Ende dieser Auswahl 493 verschiedene, nicht simulierte Datensätze zur Verfügung standen. Um die Berechnung des Benchmarks auf maximal 24 Stunden zu begrenzen, wurden 268 der 493 Datensätze aussortiert, wobei die 268 aussortierten Datensätze diejenigen waren, für die die Berechnung am längsten gedauert hätte. Für die restlichen 225 Datensätze wurde der Benchmark durchgeführt, wobei bei insgesamt 97 Datensätzen Fehler auftraten, sodass diese für die Analysen außer Acht gelassen wurden. Schlussendlich wurde also mit 128 Datensätzen gearbeitet.

4.2 Beschreibung der Datensätze

Bei den 128 Datensätzen handelt es sich ausschließlich um Datensätze, deren Response eine diskrete Variable mit endlichen Ausprägungen darstellt. Der größte Teil der Daten ($\sim 85\%$) hat dabei eine binäre Zielvariable. Der zweitgrößte Teil der Daten ($\sim 7\%$) hat einen Response mit drei Ausprägungen und die restlichen 8% der Datensätze hat einen Reponse mit mehr als drei Ausprägungen, wobei keiner der Datensätze eine Zielvariable mit mehr als 11 Ausprägungen hat. Die Anzahl der Beobachtungen in den Datensätzen ist sehr unterschiedlich und reicht von 34 bis zu 2001. Dabei beinhalten 39% der Datensätze weniger als 100 Beobachtungen und knapp 3% mehr als 1.000. Die Anzahl der Einflussvariablen reicht von 2 bis 70, wobei etwa 71% der Datensätze nicht mehr als zehn Einflussvariablen haben.

5 Analyse des Einflusses der Hyperparameter

In diesem Teil der Arbeit soll anhand der 128 Datensätze der Einfluss der Hyperparameter auf Support Vector Machines untersucht werden. Bezüglich der Hyperparameter wurden für diese jeweils vier bis fünf unterschiedliche Werte festgelegt. Wie bereits in Kapitel 3.4.2 erläutert wurde, gilt aus programmiertechnischen Gründen, je höher C, desto weniger Fehler werden bei der Modellerstellung akzeptiert und je kleiner C, umso mehr Fehler. Für den Hyperparameter C wurden insgesamt fünf verschiedene Werte ausgewählt. Dabei wurden für C sowohl hohe $(10; 5)$ als auch geringere Werte $(0.5; 0.1)$ gewählt sowie natürlich auch die Standardeinstellung mit $C = 1$. C stellt hierbei den einzigen Hyperparameter dar, der nicht Kernel spezifisch ist, also bei allen Kerneln einen Hyperparameter darstellt. Bezüglich des Hyperparameters d für den Polynomialen Kernel wurden zwei Gerade $(2; 4)$ und zwei ungerade $(3; 5)$ Werte gewählt. Bei dem Hyperparameter γ wurden vier Werte so gewählt, dass sie von 0.1 über 0.25 und 0.5, bis 1 reichen, je größer dabei γ ist, desto lokaler ist das Modell. Für die Modelle mit linearem Kernel muss pro Hyperparameter C ein Modell erstellt werden, sodass pro Datensatz fünf verschiedene Modelle untersucht werden. Für die Modelle mit polynomialem, beziehungsweise radialem Kernel werden pro Datensatz insgesamt 20 verschiedene Modelle erstellt. Diese unterscheiden sich nur in der Kombination ihrer Hyperparameter C und d, beziehungsweise C und γ. Insgesamt werden pro Datensatz also 45 unterschiedliche Modelle erstellt und die Gütemaße der Modelle (s. Kapitel 2.3) mithilfe des k-fold Verfahrens (s. Kapitel 2.2) geschätzt. Bezüglich der Untersuchung des Einflusses der Hyperparameter auf Support Vector Machines wird dabei zuerst jeder Kernel einzeln untersucht und anschließend die besten Modelle der jeweiligen Kernel miteinander verglichen. Dabei wird durchgehend die Güte der Modelle anhand des Gütemaßes multiclass Brier untersucht, da dieser noch die Sicherheit, mit der eine Beobachtung in eine Klasse zugeordnet wird, berücksichtigt und damit sensibler als beispielweiße die Accuracy ist. Wie bereits in Kapitel 2.2 erläutert ist der beste zu erreichende Wert 0, und der schlechteste Wert 1. Die in den nächsten Kapiteln getroffenen Aussagen gelten analog für die anderen Gütemaße, falls nicht explizit erwähnt werden sollte, dass dem nicht so ist. Die analogen Grafiken für die anderen Gütemaße sind dem Anhang der Arbeit beigefügt.

5.1 Support Vector Machines mit linearem Kernel

In diesem Abschnitt wird der Einfluss des Hyperparameters C auf die Güte von Support Vector Machines mit linearem Kernel untersucht. Um einen ersten Eindruck über die Verteilung des multiclass Brier und den Einfluss des Hyperparameters C zu erhalten, sind in Abbildung (11) die Boxplots des multiclass Brier, getrennt nach den Einstellungen für den Hyperparameter C, dargestellt. Dabei reicht der multiclass Brier im schlechtesten Fall von 0.862 bis hin zum besten Fall von nahezu 0. Die Boxplots der verschiedenen Modelle verhalten sich sehr ähnlich, sodass bei deren Betrachtung keine großen Unterschiede zu erkennen sind. Einzig die oberen Whisker der Modelle mit $C = 5$, beziehungsweise $C = 10$ reichen weiter als bei den restlichen Modellen. Die Mittelwerte der einzelnen Modelle, die als Punkt in den jeweiligen Boxplots dargestellt sind, sowie die Mediane der jeweiligen Modelle unterscheiden sich nur marginal. Die Mittelwerte der verschiedenen Modelle unterscheiden sich hierbei maximal um 1,35% und die Mediane um maximal 0,37%. Anhand der Grafik lässt sich nicht beurteilen, welcher Hyperparameter C zu guten Ergebnissen führt, sondern nur, dass die Güte der Modelle im Durchschnitt, beziehungsweise im Median sehr ähnlich ist.

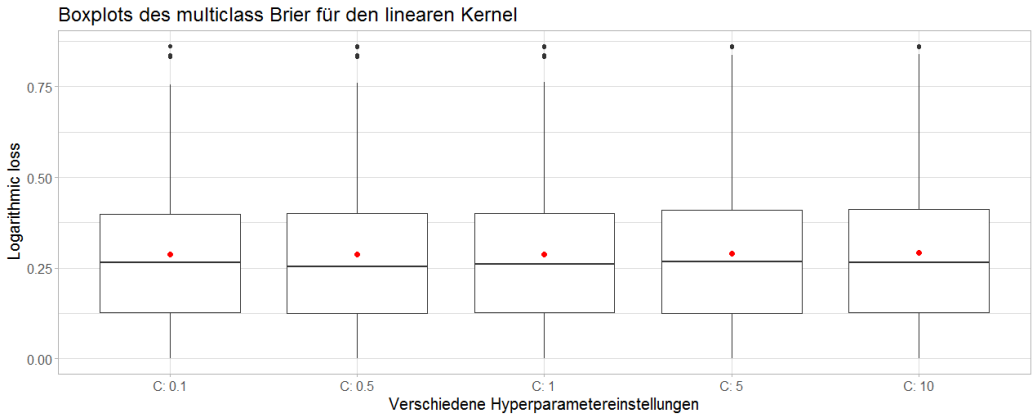


Abbildung 11: Boxplots des multiclass Brier getrennt nach den verschiedenen Einstellungen für den Hyperparameter C. Die roten Punkte stellen die Mittelwerte der jeweiligen Modelle dar.

Um zu untersuchen, wie stark sich die Güte eines Modells durch einen geeignet gewählten Hyperparameter C verbessern kann, wird nun die Differenz der jeweilgen Gütemaße von dem besten und dem schlechtesten Modell mit linearem Kernel des jeweiligen Datensatzes betrachtet. Die Differenzen dazu sind in den Boxplots links in Abbildung (12) dargestellt. Hierbei handelt es sich bei allen Gütemaßen um eine linkssteile Verteilung, sodass bei allen Gütemaßen der Median geringer als der Mittelwert ist. So beträgt beispielsweise die Differenz der Accuracy zwischen dem besten und schlechtesten Modell im Durchschnitt 2,5% und im Median nur 1,5%. Auf der einen Seite gibt es Datensätze, bei denen der Hyperparameter C die Güte der Modelle nicht zu verändern scheint, sodass sich die Gütemaße der verschiedenen Modelle überhaupt nicht unterscheiden und entsprechend bei allen Modellen gleich sind. Auf der anderen Seite gibt es bei vereinzelten Datensätze extreme Verbesserungen. So liegt beispielsweise die maximale Differenz der Accuracy bei 16,3%, sodass in solchen Fällen ein passender Hyperparameter C im Vergleich zu einem unpassend gewählten Hyperparameter C zu einer erheblichen Güteverbesserung des Modells führt.
Um die Veränderung der Güte in Abhängigkeit des Hyperparameters C zu untersuchen, wird rechts in Abbildung (12) die Differenz des multiclass Brier zwischen dem Modell mit Standardeinstellungen, also mit dem Hyperparameter $C = 1$, und den restlichen Modellen betrachtet. Hierbei wurde die Differenz über den Wert des Standardmodells minus dem Wert des entsprechenden Modelles berechnet. Folglich bedeuten Werte unter 0, dass das Standardmodell einen geringeren Wert besitzt. Im Falle des multiclass Brier als Gütemaß bedeutet dies, dass das Standardmodell eine bessere Modellgüte besitzt. Die Differenz des multiclass Brier zwischen dem Referenzmodel und den anderen Modellen liegt sowohl im Durchschnitt als auch im Median so gut wie bei 0. Die Maximale Differenz wird umso größer, je weiter der Hyperparameter C von dem Hyperparameter C des Referenzmodells entfernt ist. Dabei gehen die Differenzen sowohl in positive als auch in die negative Richtung, sodass es sowohl Datensätze zu geben scheint, bei denen ein kleiner Hyperparameter C zu einer besseren Güte des Modell führt, als auch Datensätze, bei denen die Güte des Modells mit einem kleinem Hyperparameter C wesentlich schlechter wird. Folglich kann der Hyperparameter C sowohl zu Verbesserungen als auch zu Verschlechterungen der Güte von Modellen führen.

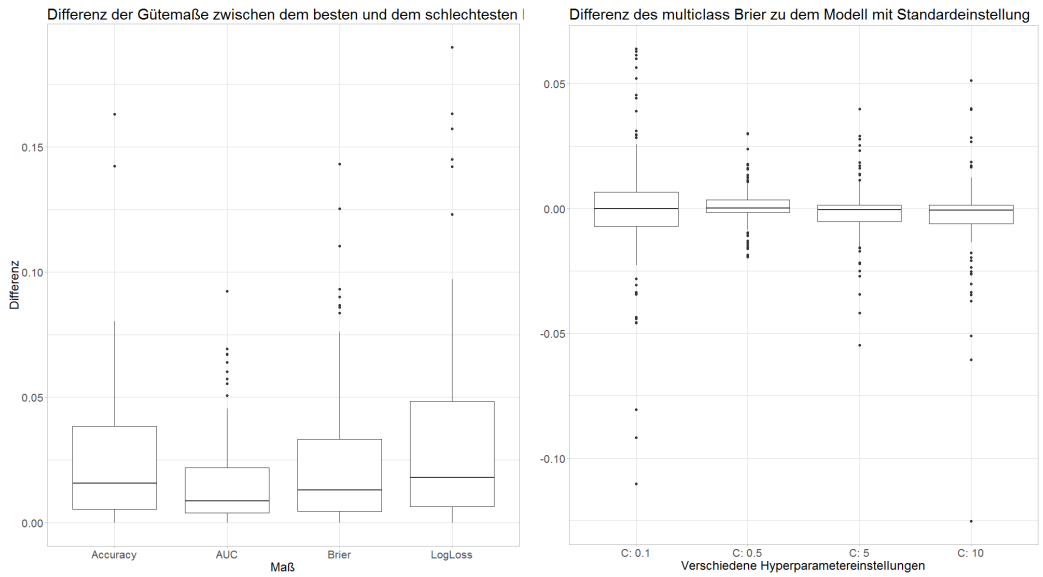


Abbildung 12: Links: Boxplots der Differenz zwischen dem besten und schlechtesten Modell, getrennt nach den Gütemaßen. Rechts: Die Differenzen des multiclass Brier der Modelle zu dem multiclass Brier des Modells mit den Standardeinstellungen (Hyperparameter $C = 1$)

Bei Betrachtung der Ränge in Abbildung (13) bestätigt sich der Eindruck, dass es Datensätze gibt, bei denen ein geringer Hyperparameter C zu einer guten Modellgüte führt und Datensätze, bei denen die gleichen Einstellungen zu einer schlechten Modellgüte führt. Es gibt keine Hyperparameter Einstellung, die immer zu einem guten Ergebniss führt. Es scheint wohl vielmehr von dem Datensatz abzuhängen, ob gewisse Einstellungen für den Hyperparameter C gute Ergebnisse erzielen oder nicht. Am besten erkennt man dies in Abbildung (13) an dem Modell mit dem Hyperparameter $C = 0.1$. Dieses Modell ist dasjenige, das mit Abstand am häufigsten den ersten Rang belegt, allerdings belegt dieses Modell auch sehr oft den schlechtesten Rang, sodass sich hier nicht pauschal sagen lässt, dass diese Einstellungen für den Hyperparameter C zu guten Ergebnissen führt. Es fällt des weiteren auf, dass das Modell mit dem Hyperparameter $C = 1$ das Modell ist, das am häufigsten auf dem dritten Rang liegt und scheint somit ein gutes Mittelmaß darzustellen. Die Modelle mit dem Hyperparameter $C = 10$ belegen am häufigsten den letzten Rang und gehören in deutlich weniger Fällen zu den Top 2 Modellen.

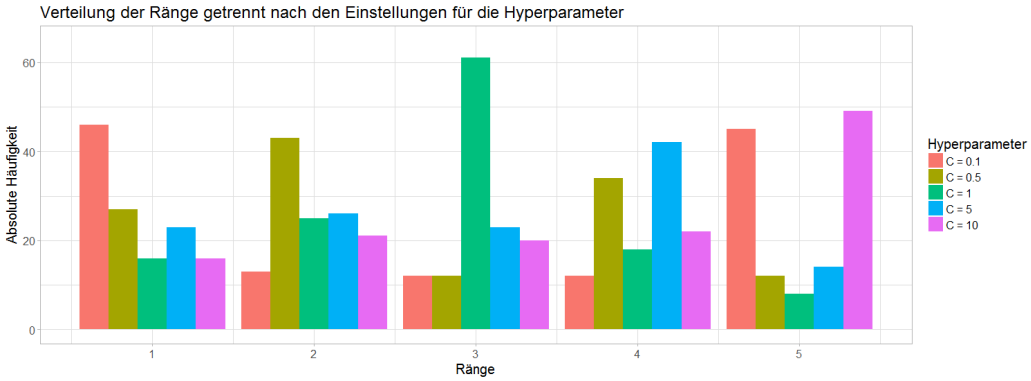


Abbildung 13: Ränge der Modelle getrennt nach den verschiedenen Hyperparameter Einstellungen. Die Ränge wurden anhand des multiclass Brier vergeben.

Um einen genaueren Überblick darüber zu erhalten, welche Hyperparameter zu guten Ergebnissen führen, wurde für jedes Gütemaß der durchschnittliche Rang eines Modells berechnet. Die Ergebnisse dessen sind in Tabelle (1) dargestellt. Was sofort ins Auge fällt ist, dass die Modelle mit $C = 10$ bei allen Gütemaßen den durchschnittlich schlechtesten Rang belegen und diese Modelle folglich im Schnitt die geringste Modellgüte besitzen. Die Modelle mit den Hyperparametern $C = 0.5$ liegen bei jedem Gütemaß im Durchschnitt auf dem ersten Platz, sodass diese Einstellung im Durchschnitt die beste für den Hyperparameter C zu sein scheint. Der durchschnittliche Rang der Modelle mit der Standardeinstellung $C = 1$ ist zwar nie der beste, belegt aber bei jedem Maß den 2. oder 3. Rang.

Maß / Hyperparameter C	$C = 0.1$	$C = 0.5$	$C = 1$	$C = 5$	$C = 10$
Accuracy	3.09	2.82	2.98	2.95	3.16
Logarithmic loss	3.00	2.67	2.95	2.93	3.44
Multiclass Brier	2.98	2.69	2.82	2.98	3.52
Multiclass AUC	2.85	2.73	2.85	3.13	3.42
Durchschnitt	2.98	2.73	2.9	2.99	3.39

Tabelle 1: Die duchschnittlich belegten Ränge der verschiedenen Modelle, getrennt nach den verschiedenen Gütemaßen. Die letzte Reihe gibt den Durchschnitt der durchschnittlich belegten Ränge an.

Zusammenfassend lässt sich sagen, dass es für den linearen Kernel keinen pauschalen Wert C gibt, der immer zu guten Ergebnissen führt. Die Modelle mit $C = 0.5$ belegen im Durchschnitt den durchschnittlich besten Rang und scheinen eine gute Einstellung für C zu sein. Die Modelle mit Hyperparameter $C = 10$ belegen durchschnittlich den schlechtesten Rang, sodass höhere Werte wohl tendenziell schlechter funktionieren als Modelle mit geringeren Werten für C.

5.2 Support Vector Machines mit polynomialen Kernel

In diesem Abschnitt der Arbeit wird der Einfluss der Hyperparameter C und d auf die Güte von Support Vector Machines mit polynomialem Kernel untersucht. Für eine erste Übersicht sind in Abbildung (14) die Boxplots des multiclass Brier getrennt nach den verschiedenen Einstellungen für die Hyperparameter dargestellt. Die gelben Punkte innerhalb der Boxplots stellen die Mittelwerte der entsprechenden Modelle dar. Die gestrichelten grauen Linien zwischen den Boxplots teilen die Modelle nach dem Hyperparameter d auf, sodass beispielsweise alle Modelle vor der ersten gestrichelten Linie den Hyperparameter $d = 2$ haben und sich so nur in dem Hyperparameter C unterscheiden. Die Boxplots der Modelle mit dem gleichen Hyperparameter C sind in der gleichen Farbe dargestellt. Bei Betrachtung der Modelle mit dem gleichen Hyperparameter d lässt sich erkennen, dass die Modelle trotz unterschiedlichem Hyperparameter C eine sehr ähnliche Verteilung haben. Folglich unterscheiden sich die Mittelwerte und Mediane der Modelle mit gleichem Hyperparameter d nicht besonders stark. Die Abweichung der Mittelwerte betrug maximal 1.3% und die der Medianen maximal 1.77%. Bei Betrachtung der Modelle die sich nur in dem Hyperparameter d unterscheiden, also den gleichen Hyperparameter C besitzen, scheinen die Unterschiede deutlich größer zu sein. So lässt sich bereits anhand der Boxplots erkennen, dass Modelle mit $d = 3$ sowohl im Durchschnitt als auch im Median eine bessere Modellgüte besitzen als die entsprechenden Modelle mit $d = 4$. Folglich weichen Mittelwerte und Mediane deutlich stärker voneinander ab, sodass die Differenz der Mediane bei maximal 3.33% liegt und damit etwa doppelt so hoch wie die maximale Differenz der Mediane zwischen den Modellen mit gleichem Hyperparameter d. Die maximale Differenz der Mittelwerte beträgt etwa 6.69%, was knapp 4.5 mal so hoch ist wie die maximale Differenz der Mittelwerte zwischen den Modellen mit gleichem Hyperparameter d. Diesen Ergebnissen nach zu urteilen, hat der Hyperparameter d einen stärkeren Einfluss auf die durchschnittliche Modellgüte als der Hyperparameter C.

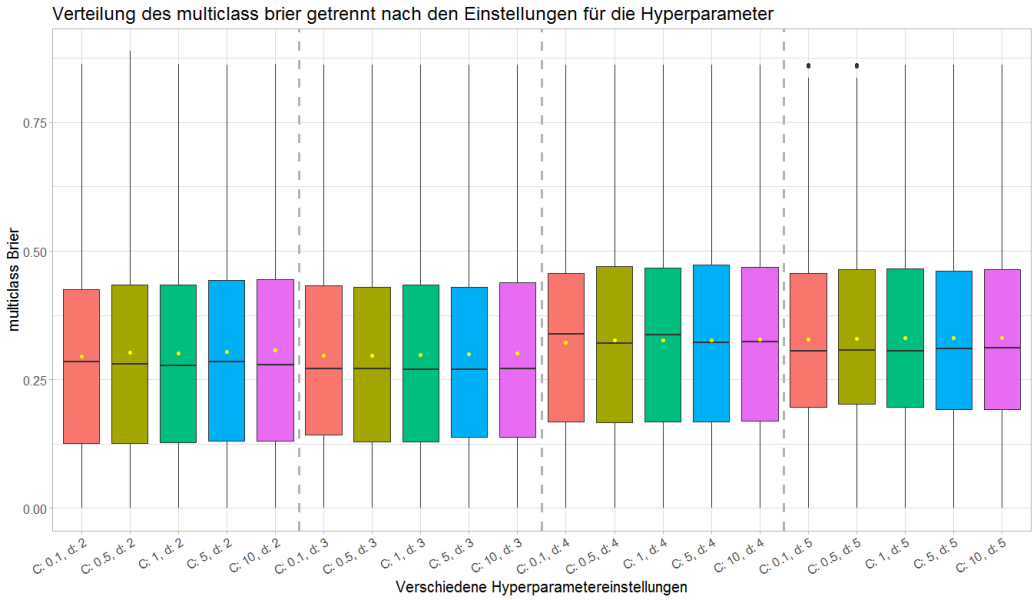


Abbildung 14: Boxplots des multiclass Brier getrennt nach den verschiedenen Einstellungen für die Hyperparameter C und d. Die gelben Punkte stellen die Mittelwerte der jeweiligen Modelle dar.

Um den Einfluss der Hyperparameter d und C auf die Modellgüte zu untersuchen, sind in Abbildung (15) die Differenzen des mutliclass Brier zwischen dem Modell mit den Standardeinstellungen $C = 1$ und $d = 3$ dargestellt. Wie bereits in Kapitel 5.1 erläutert, bedeuten Werte kleiner als 0, dass das Standardmodell eine bessere Modellgüte besitzt. Links in Abbildung (15) sind die Differenzen der Modelle veranschaulicht, die sich im Vergleich zu dem Standardmodell nur in dem Hyperparameter C unterscheiden. Hierbei erkennt man, dass im größten Teil der Fälle der Hyperparameter C nur einen geringen Einfluss auf die Modellgüte zu haben scheint. Die Abweichung von dem mutliclass Brier des Standardmodells beträgt bei allen Modellen sowohl im Median als auch im Mittelwert nie mehr als 0.005. Auch die Quartile der Boxplots liegen sehr nahe an 0. Dennoch gibt es immer wieder vereinzelnd Fälle, bei denen die Abweichung zu dem Standardmodell sehr groß ist. Der Hyperparameter C scheint also in den meisten Fällen zu keinen großen Veränderungen der Modellgüte beizutragen, kann in vereinzelten Fällen aber die Güte eines Modells stark verändern. Rechts in Abbildung (15) sind die Differenzen der Modelle dargestellt, die sich im Vergleich zu dem Standardmodell nur in dem Hyperparameter d unterscheiden. Hierbei erkennt man, dass der Hyperparameter d wohl einen stärkeren Einfluss zu haben scheint, da die Streuung der Daten wesentlich größer ist. So ist es auch nicht ver-

wunderlich, dass sich die Modelle im Vergleich zu dem Standardmodell im Mittelwert um bis zu 0.06 und im Median um bis zu 0.03 unterscheiden. Der Hyperparameter d scheint im Mittel also einen wesentlich stärkeren Einfluss auf die Güte der Modelle zu haben, als der Hyperparameter C. Auch gibt es wesentlich mehr Ausreißer als bei den Modellen, die sich nur in dem Hyperparameter C unterscheiden. Desweiteren fällt auf, dass die Modelle mit $d = 4$, beziehungsweise $d = 5$ in über 75% der Fälle eine schlechtere Modellgüte als das Standardmodell haben, da deren drittes Quartil negativ ist.

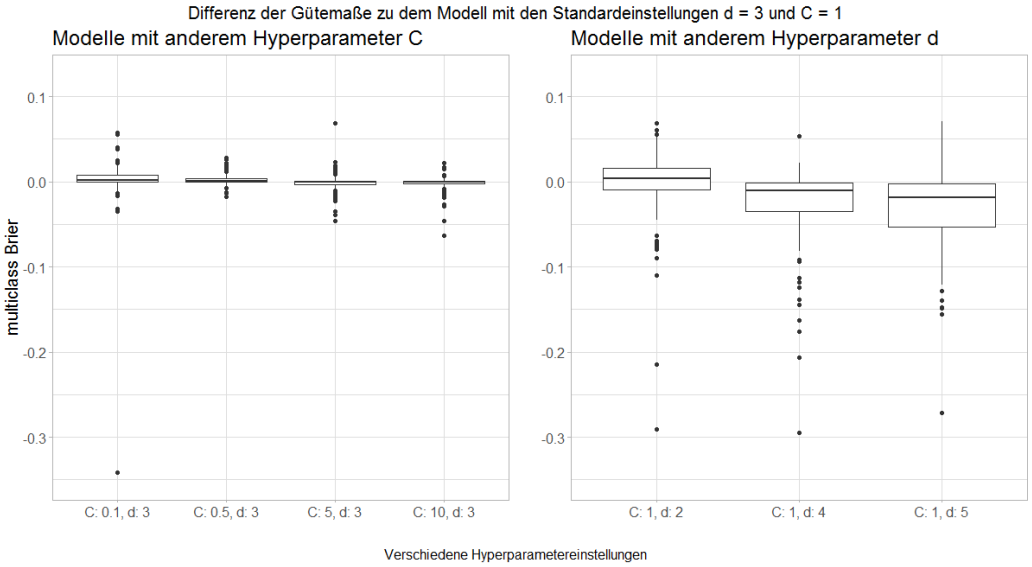


Abbildung 15: Boxplots der Differenzen des multiclass Brier zu dem Modell mit den Standardeinstellungen $C = 1$ und $d = 3$. Links: Modelle, die sich zu dem Standardmodell nur im Hyperparameter C unterscheiden. Rechts: Modelle, die sich zu dem Standardmodell nur im Hyperparameter d unterscheiden.

Wie bereits festgestellt wurde, scheint der Hyperparameter d einen wesentlich größeren Einfluss auf die Modellgüte zu haben als der Hyperparameter C. Zudem wurde festgestellt, dass die Modelle mit $d = 4$ beziehungsweise $d = 5$ in über 75% der Fälle schlechter abschneiden als das Standardmodell. Diese Feststellungen bestätigen sich bei Betrachtung der Ränge in Abbildung (16). Obwohl diese wegen der 20 verschiedenen Modelle ein wenig überladen ist, sieht man recht schnell, dass die Modelle mit $d = 2$ oder $d = 3$ mit Abstand am häufigsten unter den Top 9 Rängen vertreten sind. Zudem fallen schnell die Modelle auf, die am häufigsten die schlechtesten Ränge belegen. Bei diesen Modellen handelt es sich so gut wie ausschließlich um die Modelle mit dem Hyperparameter $d = 5$. Auch die Modelle mit $d = 4$ schneiden

wesentlich schlechter ab, als die Modelle mit $d = 2$ beziehungsweise $d = 3$. Bezüglich des Hyperparameters C lässt sich anhand dieser Grafik keine Aussage treffen, da der belegte Rang stärker mit dem Hyperparameter d zusammenzuhängen scheint.

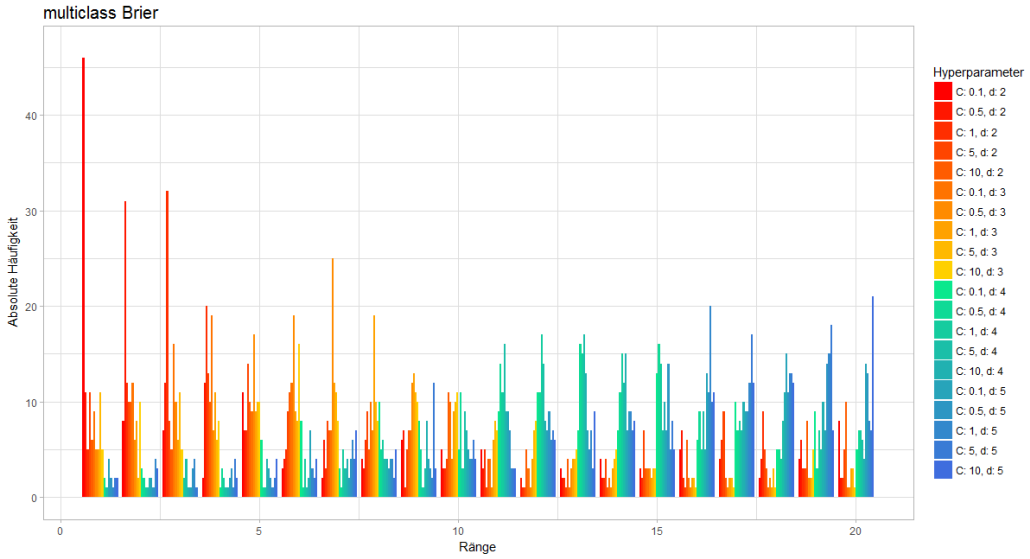


Abbildung 16: Ränge der verschiedenen Modelle anahand des multiclass Brier

Um den Einfluss der einzelnen Hyperparameter d und C auf die Modelle besser beurteilen zu können, sind in Abbildung (17) die Ränge der Modelle dargestellt, wobei bei den Modellen jeweils einer der zwei Hyperparameter fixiert wurde. In der oberen Grafik sind nur die Modelle mit Standard Hyperparameter $d = 3$ dargestellt, sodass sich die Modelle nur in ihrem C unterscheiden. Hierbei lässt sich schnell erkennen, dass die Ränge der Modelle tendenziell umso schlechter werden, je höher der Hyperparameter C ist. In der unteren Grafik sind nur die Modelle mit Standard Hyperparameter $C = 1$ dargestellt, sodass sich die Modelle folglich nur in ihrem Hyperparameter d unterscheiden. Hierbei sieht man klar, dass die Modelle mit geringerem Hyperparameter d wesentlich besser abschneiden als die Modelle mit höher gewähltem d. Die soeben getroffenen Aussagen gelten komplett analog, wenn die fixierten Hyperparameter andere Werte annehmen. Allerdings ist der Einfluss des Hyperparameters C nicht mehr ganz so deutlich, wenn d auf 4 oder 5 fixiert wird. Die entsprechenden Grafiken hierfür befinden sich im Anhang.

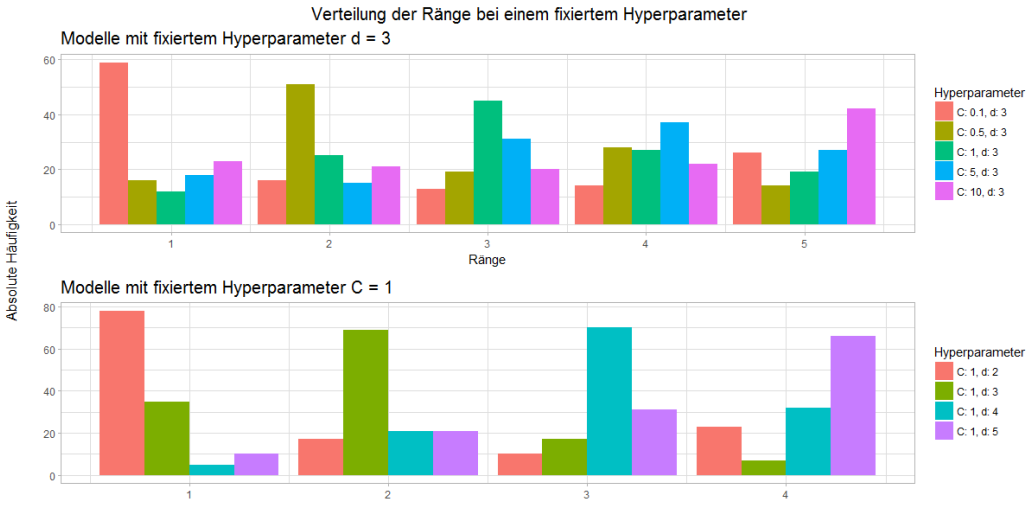


Abbildung 17: Ränge der Modelle anhand des multiclass Brier, wobei jeweils einer der zwei Hyperparameter d oder C der Modelle fixiert ist.

Tabelle (2) stellt die durchschnittlich belgten Ränge der Modelle für das Gütemaß multiclass Brier dar. Die Modelle mit $d = 2$ sind diejenigen Modelle, die fast immer den besten Platz belegen, wenn man sie mit den Modellen vergleicht, die den gleichen Hyperparameter C haben. Der Durchschnitt der durchschnittlich belegte Ränge ist für die Modelle mit $d = 3$ am besten. Es fällt zudem auf, dass der durchschnittliche Rang, der belegt wird, eher mit dem Hyperparameter d als mit dem Hyperparameter C zusammenhängt. Erkennen lässt sich dies daran, dass die Modelle mit geeignet gewähltem Hyperparameter d (2; 3) unabhängig von deren Hyperparameter C einen durchschnittlich besseren Rang belegen als die Modelle mit ungeeignet gewähltem d (4; 5). Bezüglich des Hyperparameters C erkennt man, dass bei den Modellen mit gleichem d fast immer die Modelle besser abschneiden die einen geringen Hyperparameter C haben. Erkennen lässt sich dies auch daran, dass der Durchschnitt der durchschnittlich belegten Ränge umso schlechter wird, je höher C ist.

Hyperparameter: C / d	$d = 2$	$d = 3$	$d = 4$	$d = 5$	Durchschnitt
$C = 0.1$	6.72	6.27	12.15	13.45	9.65
$C = 0.5$	6.69	7.04	13.02	14.07	10.21
$C = 1$	7.20	7.87	13.03	13.91	10.50
$C = 5$	8.26	8.35	12.93	14.02	10.89
$C = 10$	9.45	8.26	13.18	13.56	11.11
Durchschnitt	7.66	7.56	12.86	13.80	

Tabelle 2: Die duchschnittlich belegten Ränge der verschiedenen Modelle für den multiclass Brier. Die letzte Zeile gibt den Durchschnitt der durchschnittlich belegten Ränge für die Modelle mit Hyperparameter d an. Die äußerste Spalte gibt den Durchschnitt der durchschnittlich belegten Ränge für die Modelle mit Hyperparameter C an

Zusammenfassend lässt sich festhalten, dass bei Support Vector Machines mit polynomialem Kernel der Einfluss des Hyperparameters d deutlich größer als der des Hyperparameters C ist. Dabei schneiden die Modelle mit geringerem Hyperparameter d deutlich besser ab als die Modelle mit einem höheren d. Ein Grund hierfür könnte sein, dass bei einem hohen d der Merkmalsraum zu stark vergrößert wird, was zu einer Überanpassung der Modelle führt. Der Hyperparameter C hat einen wesentlich geringeren Einfluss, kann die Modellgüte in einigen Fällen dennoch stark beeinflussen. Dabei haben Modelle mit geringerem C im Durchschnitt eine bessere Modellgüte vorzuweisen.

5.3 Support Vector Machines mit radialem Kernel

Im folgenden soll nun der Einfluss der Hyperparameter C und γ auf die Güte von Support Vector Machines mit polynomialem Kernel untersucht werden. Für eine erste Übersicht sind in Abbildung (18) die Boxplots des multiclass Brier getrennt nach den verschiedenen Einstellungen für die Hyperparameter dargestellt. Die gelben Punkte innerhalb der Boxplots stellen die Mittelwerte der entsprechenden Modelle dar und die Modelle mit dem gleichen Hyperparameter C sind in der gleichen Farbe dargestellt. Die gestrichelten grauen Linien zwischen den Boxplots teilen die Modelle nach dem Hyperparameter γ auf, sodass beispielsweise alle Modelle vor der ersten gestichelten Linie den Hyperparameter $\gamma = 0.1$ haben und sich so nur in dem Hyperparameter C unterscheiden. Innerhalb der Modelle mit dem gleichen Hyperparameter γ weisen die Verteilungen eine klare Struktur auf. Je kleiner der Hyperparameter C ist, desto höher liegt der mutliclass Brier und umso schlechter ist entsprechend die Modellgüte. So liegen folglich auch die Mediane, beziehungs-

weise die Mittelwerte entsprechend höher, je kleiner C ist, sodass Modelle mit größerem Hyperparameter C im Durchschnitt, beziehungsweise Median eine bessere Modellgüte als Modelle mit gleichem γ aber geringerem C aufweisen. Bei Betrachtung der Modelle, die sich nur in dem Hyperparameter γ unterscheiden, also den gleichen C Wert haben, lässt sich erkennen, dass der Mittelwert, beziehungsweise Median des mutliclass Brier umso kleiner ist, je geringer der Hyperparameter γ ist. Modelle mit geringerem Hyperparameter γ weisen dementsprechend eine besser Modellgüte auf als Modelle mit gleichem C, aber höherem γ. Anhand der Verteilung des multiclass Brier entsteht also der Eindruck, dass die Modelle bei kleinem γ und gleichzeitig großem C die durchschnittlich beste Güte besitzen.

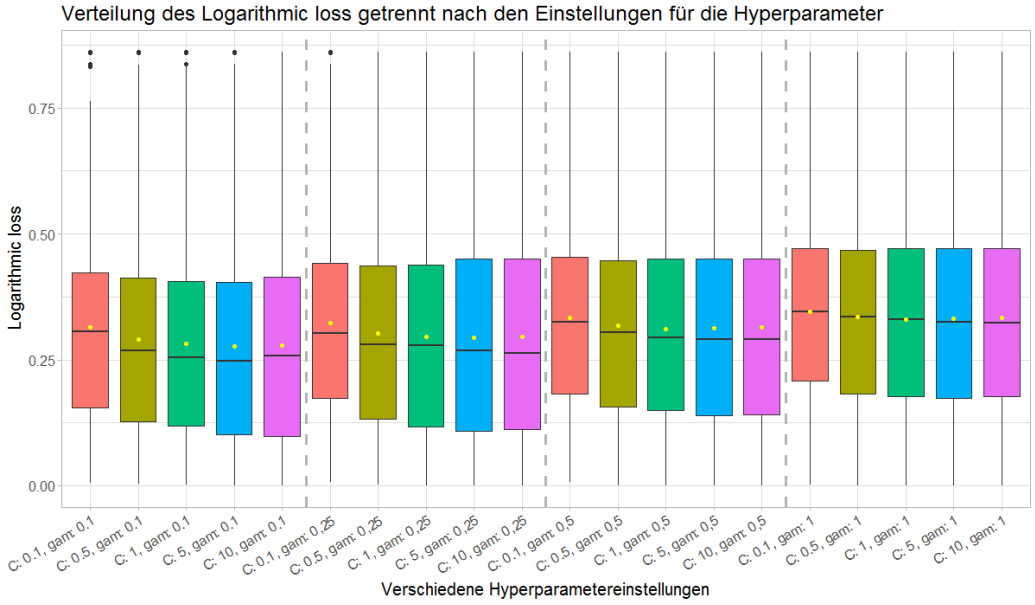


Abbildung 18: Boxplots des multiclass Brier getrennt nach den verschiedenen Einstellungen für die Hyperparameter C und γ. Die gelben Punkte stellen die Mittelwerte der jeweiligen Modelle dar.

Um den Einfluss der einzelnen Hyperparameter auf die Modellgüte zu untersuchen, werden nun die Differenzen der Modelle zu dem Standardmodell mit $C = 1$ und $\gamma = 1$ betrachtet. Links in Abbildung (19) sind die Differenzen des multiclass Brier zwischen dem Standardmodell und den Modellen mit anderem Hyperparameter C dargestellt. Wie bereits in Kapitel 5.1 erläutert, bedeuten Werte kleiner als 0, dass das Standardmodell eine bessere Modellgüte besitzt. Bei all diesen Modellen beträgt die Abweichung im Mittel beziehungsweise Median so gut wie 0. Die Modelle mit $C = 0.1$ schneiden

hierbei am schlechtesten ab, da bei diesen die meisten Datenpunkte unter 0 liegen und die Ausreißer nur in negative Richtung gehen. Folglich führen die Modelle mit $C = 0.1$ gegenüber dem Standardmodell eher zu einer Verschlechterung der Modellgüte. Die Abweichungen von dem Standardmodell sind nur in vereinzelnden Fällen wirklich groß, sodass der Eindruck entsteht, dass der Hyperparameter C tendenziell keinen besonders großen Einfluss zu haben scheint. In vereinzelnden Datensätzen kann der Hyperparameter C allerdings auch zu erheblichen Veränderungen der Modellgüte führen.
Rechts in der Abbildung sind die Differenzen des multiclass Brier zwischen dem Standardmodell und den Modellen mit anderem Hyperparameter γ dargestellt. Hierbei sieht man, dass die Streuung der Daten wesentlich größer ist als bei den Modellen, die sich nur in dem Hyperparameter C unterscheiden. Der Einfluss von γ scheint wesentlich stärker zu sein als der von C. Der Eindruck, dass die Modellgüte im Mittel besser ist, umso geringer γ ist, bestätigt sich bei Betrachtung der Differenzen zu dem Standardmodell, da bei allen Modellen das erste Quartil über 0 liegt. Im Vergleich zu dem Modell mit $\gamma = 1$ verbessert sich die Modellgüte bei 75% der Datensätze, wenn für den Hyperparameter γ der Wert 0.1, 0.25 oder 0.5 gewählt wird. Die maximale Veränderung der Modellgüte ist dabei umso größer, je kleiner γ ist.

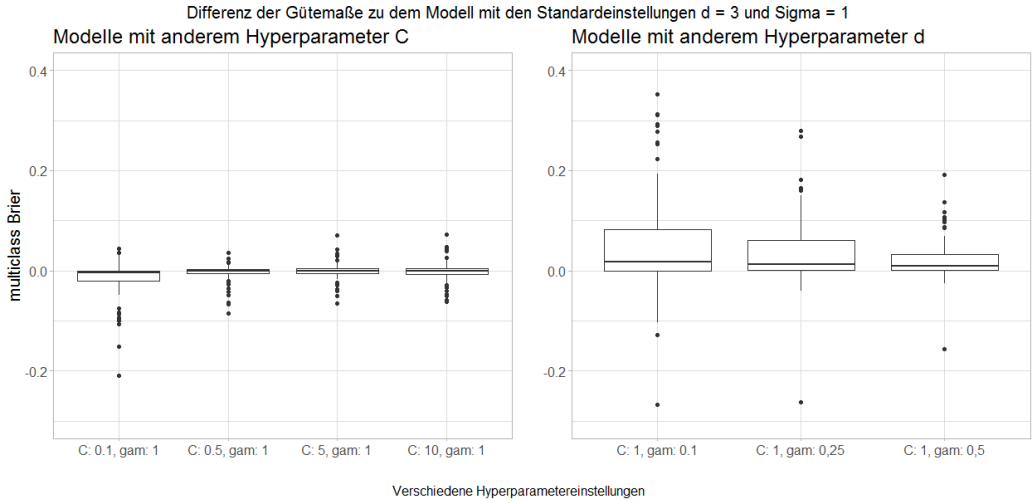


Abbildung 19: Boxplots der Differenzen des mutliclass Brier zu dem Modell mit den Standardeinstellungen $C = 1$ und $\gamma = 1$. Links: Modelle, die sich zu dem Standardmodell nur im Hyperparameter C unterscheiden. Rechts: Modelle, die sich zu dem Standardmodell nur im Hyperparameter γ unterscheiden.

Wie bereits anhand der vorherigen Grafiken untersucht wurde, scheint der Kernelspezifische Hyperparameter γ einen deutlich höheren Einfluss auf die Modellgüte zu haben als der Hyperparameter C. In Abbildung (20) sind die Ränge, die anhand des mutlticlass Brier gebildet wurden, dargestellt. Hierbei erkennt man, dass die Ränge, die die Modelle belegen, tendenziell umso besser sind, je kleiner γ ist und umso schlechter, je größer. So belegen die Modelle mit $\gamma = 0.1$ oder $\gamma = 0.25$ entsprechend auch am häufigsten die Ränge 1 bis 9. Die Modelle mit höherem γ sind am häufigsten auf den schlechteren Rängen. Allerdings gibt es auch vereinzelnd Ausnahmen, wie beispielsweise das Modell mit den Hyperparametern $C = 0.1$ und $\gamma = 0.1$, das am zweit häufigsten den letzten Rang belegt, obwohl dieses Modell einen sehr niedrigen γ Wert hat. Bezüglich des Hyperparameters C lässt sich anhand dieser Grafik keine pauschale Aussage treffen, da die Ränge der Modelle wohl stärker mit γ zusammenhängen zu scheinen als mit C.

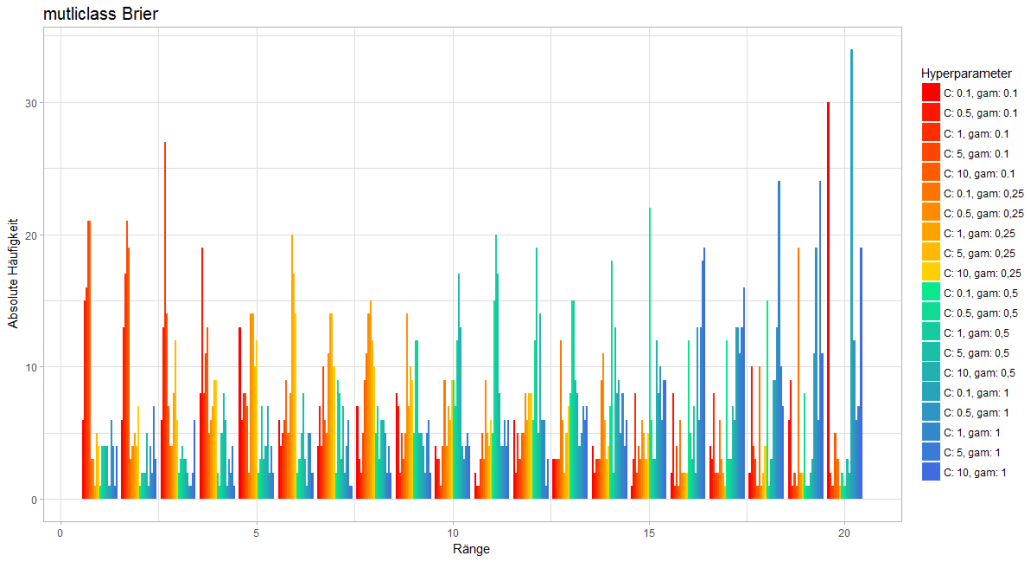


Abbildung 20: Ränge der verschiedenen Modelle anahand des multiclass Brier

Um den Einfluss der Hyperparameter auf die Ränge der Modelle gesondert zu untersuchen, werden nun die Ränge der Modelle betrachtet, wobei bei den Modellen jeweils einer der 2 Hyperparameter fixiert wurde. Die Ergebnisse sind in Abbildung (21) dargestellt. In der oberen Grafik wurde der Hyperparameter γ auf 1 fixiert, sodass sich die Modelle nur in ihrem Hyperparameter C unterscheiden. Hierbei scheint es keine klare Hierarchie zu geben, sodass der erste Rang von allen Modellen nahezu gleich oft belgt wurde. Die Modelle mit kleinem C scheinen jedoch besonders oft unter den letzten Rängen zu

sein, sodass die Modellgüte bei einem gering gewählten C tendenziell eher schlechter wird. Des weiteren fällt noch auf, dass die Modelle mit $C = 1$ mit Abstand am häufigsten den dritten Rang belegen und mit Abstand am seltensten auf den vierten und fünften Rängen liegen, sodass radiale Support Vector Machines mit $C = 1$ gut zu funktionieren scheinen. In dem unteren Teil der Abbildung sind die Ränge der Modelle mit fixiertem Hyperparameter C dargestellt, sodass sich die Modelle nur in ihrem Hyperparameter γ unterscheiden. Hierbei ist eine klare Hierarchie zu erkennen. Das Modell mit dem geringsten Hyperparameter γ ist am häufigsten auf dem ersten Rang vertreten und das Modell mit dem höchsten Hyperparameter γ am häufigsten auf dem letzten Rang. Dies bestätigt den Eindruck aus Abbildung (20), dass die Modelle umso besser abschneiden, je geringer γ ist und das bezüglich des Hyperparameters C keine pauschale Aussage getroffen werden kann. Die soeben getroffenen Aussagen gelten komplett analog, wenn die fixierten Hyperparameter andere Werte annehmen. Die entsprechenden Grafiken hierfür befinden sich im Anhang.

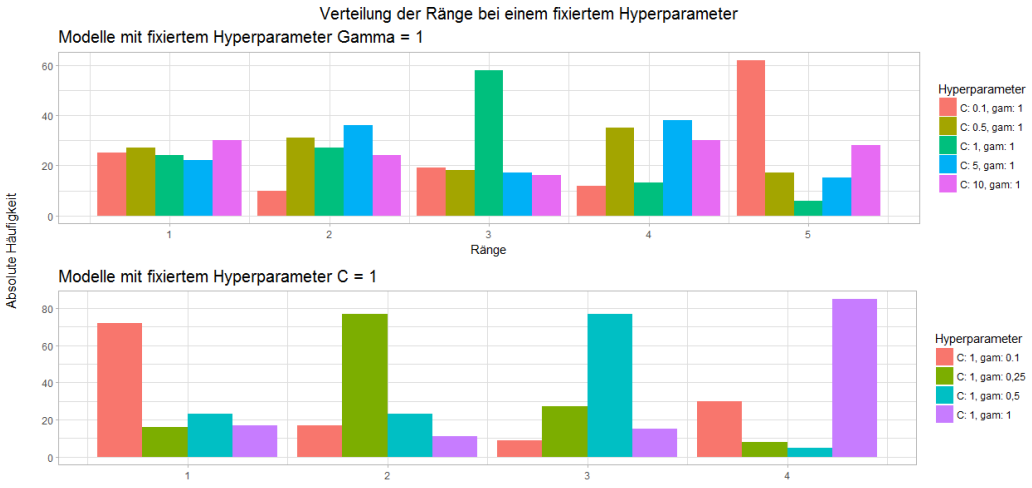


Abbildung 21: Ränge der Modelle anhand des multiclass Brier, wobei jeweils einer der zwei Hyperparameter d oder C der Modelle fixiert ist.

Tabelle (3) stellt die durchschnittlich belgten Ränge der Modelle für das Gütemaß multiclass Brier dar. Hierbei bestätigt sich der Eindruck, dass die Modelle umso besser abschneiden, je geringer der Hyperparameter γ ist. Die Modelle mit $\gamma = 0.1$ sind diejenigen, die immer den besten Platz belegen, wenn man sie mit den Modellen vergleicht, die den gleichen Hyperparameter C haben. Die durchschnittlich belegten Ränge werden dabei mit steigendem Hyperparameter γ schlechter, folglich ist der Durchschnitt der durchschnitt-

lich belegten Ränge umso niedriger, je geringer γ ist. Bei dem Hyperparameter C fällt sofort auf, dass die Modelle mit sehr geringem C (0.1; 0.5) immer die schlechtesten durchschnittlichen Ränge belegen, wenn man sie mit den Modellen vergleicht, die sich nur in ihrem Hyperparameter d unterscheiden, sodass die Modellgüte unter geringeren Werten für C zu leiden scheint. Vergleicht man die durchschnittlich belegten Plätze der Modelle mit gleichem d, so belegen die Modelle mit einem höheren Hyperparameter C (1; 5) immer die besten durchschnittlichen Ränge. Bei diesen tut sich allerdings noch der Wert $C = 5$ hervor, dessen Durchschnitt der durchschnittlichen Ränge am besten ist.

Hyperparameter: C / γ	$\gamma = 0.1$	$\gamma = 0.25$	$\gamma = 0.5$	$\gamma = 1$	Durchschnitt
$C = 0.1$	11.09	12.17	13.63	15.36	13.06
$C = 0.5$	7.92	8.89	10.82	13.57	10.30
$C = 1$	6.87	7.97	9.74	13.31	9.47
$C = 5$	6.09	7.64	10.70	14.06	9.62
$C = 10$	6.60	8.30	10.95	14.28	10.03
Durchschnitt	7.71	8.99	11.16	14.12	

Tabelle 3: Die duchschnittlich belegten Ränge der verschiedenen Modelle für den mutliclass Brier. Die letzte Zeile gibt den durchschnittlichen Rang für das Modell mit Hyperparameter d an. Die äußerste Spalte gibt den durchschnittlichen Rang für das Modell mit Hyperparameter C an

Zusammenfassend lässt sich sagen, dass bei Support Vector Machines mit radialem Kernel Kernel der Hyperparameter γ einen höheren Einfluss auf die Modellgüte hat als der Hyperparameter C. Modelle mit geringer gewählten γ weisen dabei im Durchschnitt eine bessere Modellgüte auf. Modelle mit höherem γ schneiden deutlich schlechter ab, was eventuell darauf zurückzuführen ist, dass diese Modelle zu lokal wirken und dies eine Überanpassung an die Daten zur Folge haben kann. Der Hyperparameter C hat einen geringeren Einfluss auf die Modellgüte, kann in vereinzelten Fällen die Güte eines Modells aber stark beeinflussen. Modelle mit kleinem Hyperparameter C (0.1; 0.5) schneiden dabei durchschnittlich am schlechtesten ab. Die Modelle mit $C = 1$ und $C = 5$ führen zu den durchschnittlich besten Modellgüten.

5.4 Vergleich der besten Modelle, der jeweiligen Kernel

Um einen Eindruck darüber zu bekommen, welche Kernel zu guten und welche zu eher schlechten Modellgüten führen, werden in diesem Teil der Arbeit die besten Modelle der jeweilgen Kernel miteinander verglichen. Hierbei wird als Maßstab für die Güte der Modelle der durchschnittlich belegte Rang bei dem Gütemaß multiclass Brier verwendet. Bei den Modellen mit linearem Kernel war das Modell mit $C = 0.5$ das Modell, das den durchschnittlich besten Rang belegt hat, bei dem polynomialen Kernel war es das Modell mit $d = 3$ und $C = 0.1$ und bei den Modellen mit radialem Kernel war das Modell mit $\gamma = 0.1$ und $C = 5$ das Modell mit dem durchschnittlich besten Rang.
Zur Betrachtung der Verteilung des multiclass Brier der verschiedenen Modelle soll Abbildung (22) dienen. Bei Betrachtung der Verteilung lässt sich erkennen, dass die Verteilung der verschiedenen Modelle nicht besonders unterschiedlich ist. Es sind nur leichte Differenzen zu erkennen. Dabei scheint die Güte der Modelle mit radialem Kernel am besten zu sein, da dort sowohl der Durchschnitt als auch der Median den geringsten Wert hat. Bei den Modellen mit polynomialem Kernel ist die Modellgüte im Schnitt am schlechtesten, sodass entsprechend der Mittelwert beziehungsweise Median etwa um 0.02 höher liegt als bei den Modellen mit radialem Kernel.

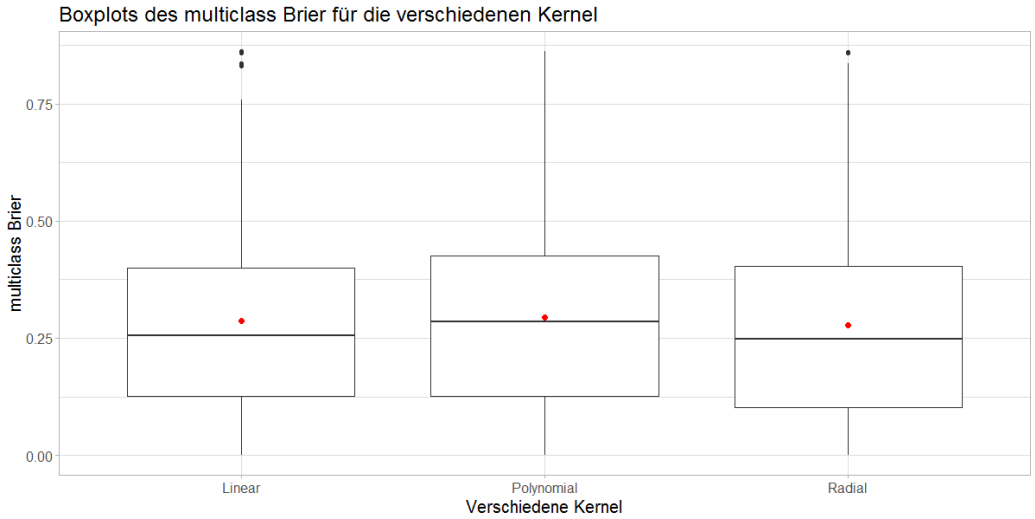


Abbildung 22: Boxplots des multiclass Brier getrennt nach den verschiedenen Kerneln. Die roten Punkte stellen die Mittelwerte der jeweiligen Modelle dar.

Die Differenzen des multiclass Brier der Modelle mit den verschiedenen Kerneln sind in Abbildung (23) dargestellt. Alle Werte, die größer 0 sind, bedeuten, dass das Modell mit dem zweitgenannten Kernel eine bessere Modellgüte besitzt, da der Wert des erstgenannten Modells einen höheren multiclass Brier besitzt und damit eine schlechtere Modellgüte. Bei dem Vergleich der Modelle mit linearem Kernel mit den Modellen mit polynomialen Kernel erkennt man, dass bei über 50% der Datensätze Modelle mit polynomialem Kernel eine bessere Modellgüte vorzuweisen haben, da der Median im positiven Bereich liegt. Bei dem Vergleich mit dem radialen Kernel liegt der Median im negativen Bereich, weshalb die linearen Modelle bei über 50% der Datensätze eine bessere Modellgüte vorzuweisen haben. Die Differenzen zwischen den Modellen mit linearen Kernel und dem radialen beziehungsweise polynomialen Kernel reichen in beiden Fällen bis zu 0.85, was eine enorme Veränderung der Modellgüte bedeutet. Hierbei kann ein lineares Modell sowohl zu erheblichen Verbesserungen als auch zu erheblichen Verschlechterungen führen. Bei Betrachtung der Differenzen zwischen den Modellen mit radialem Kernel mit den Modellen mit polynomialem Kernel erkennt man, dass die Streuung hier wesentlich kleiner ist, sodass die Modellgüte sich um maximal 0.19 verändert. Dabei liegt das dritte Quartil des entsprechenden Boxplots unter 0, sodass folglich bei über 75% der Datensätze die Modellgüte besser ist, wenn der Kernel ein radialer und kein polynomialer ist.

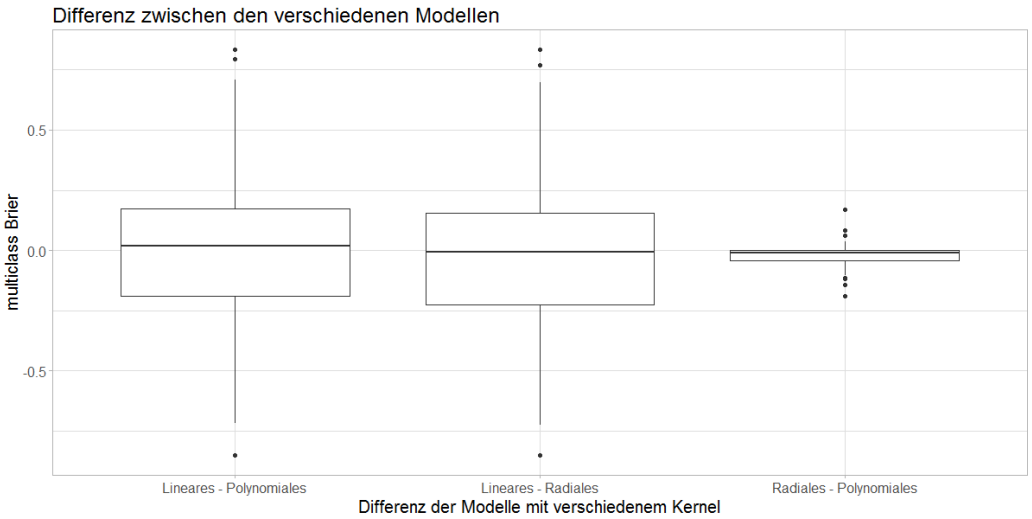


Abbildung 23: Boxplots der Differenzen des multiclass Brier getrennt nach den verschiedenen Kerneln.

In Abbildung (24) sind die Ränge der Modelle dargestellt. Die Modelle mit radialem Kernel schneiden deutlich am besten ab und belegen am häufigsten den ersten Rang und am seltensten den letzten Rang. Bei den Modellen mit polynomialem Kernel ist dies genau umgekehrt. Bei den Modellen mit linearem Kernel ist dies nicht so klar, da diese zwar am zweithäufigsten den ersten Rang belegen, allerdings auch am zweithäufigsten den letzten. Bei diesen Modellen scheint es wohl sehr abhängig vom Datensatz zu sein, wie gut die Modellgüte sein wird. Bei den durchschnittlich belegten Rängen ist das Modell mit radialem Kernel bei den Gütemaßen AUC, multiclass Brier und logarithmic loss am besten. Die Modelle mit polynomialem Kernel belegen bei allen Gütemaßen den durchschnittlich schlechtesten Rang.

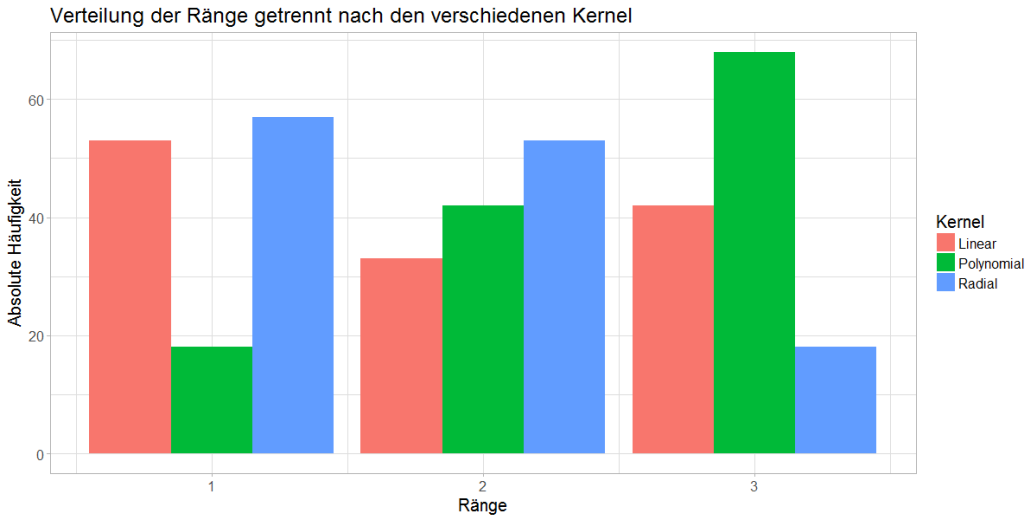


Abbildung 24: Ränge der Modelle getrennt nach dem Kernel anhand des Logarithmic loss.

Zusammenfassend lässt sich sagen, dass die Modelle mit radialem Kernel in den meisten Fällen den Modellen mit polynomialem Kernel überlegen sind. Die Differenz der Modellgüte ist dabei zwischen den Modellen mit radialem und polynomialem Kernel am geringsten. Die Abweichungen der Gütemaße zu den Modellen mit linearem Kernel ist dabei wesentlich größer. Modelle mit linearem Kernel erzielen zwar auch gute Ergebnisse, schaffen es aber nicht, konstant bessere Modelle zu erstellen als z.B. Modelle mit radialem Kernel, sodass es bei diesen sehr abhängig vom Datensatz zu sein scheint.

6 Fazit und Ausblick

Die verschiedenen Einstellungsmöglichkeiten für die Hyperparameter von Support Vector Machines haben direkten Einfluss auf die Modelle und somit auf die Modellgüte. Dabei können die Unterschiede der Modellgüte teilweise sehr gravierend sein. Bei den Modellen mit linearem Kernel wurde der Einfluss des Hyperparameters C untersucht. Dabei hat sich gezeigt, dass sehr hohe oder sehr kleine Werte zu eher schlechteren Modellen führen und Modelle mit $C = 0.5$, beziehungsweise $C = 1$ die besseren Ergebnisse liefern. Bei den Modellen mit polynomialem, beziehungsweise radialem Kernel wurde neben dem Einfluss des Hyperparameters C auch noch der der Kernelspezifischen Hyperparameter d, beziehungsweise γ untersucht. In beiden Fällen war der Einfluss des kernelspezifischen Hyperparameter wesentlich größer als der des Hyperparameters C, sodass dieser eher eine untergeordnete Rolle bei diesen Kerneln spielt. Bei dem polynomialen Kernel war die Modellgüte umso besser, je geringer der Hyperparameter C war und entsprechend umso schlechter, je höher C gewählt wurde. Bezüglich des Hyperparameters d ist die Modellgüte am besten, wenn dieser nicht größer als 3 gewählt wird, da dies unter Umständen zu einer Überanpassung führt und die Modellgüte folglich stark nachlässt. Bei den Modellen mit radialem Kernel haben die Modelle mit einem kleinen Hyperparameter C wesentlich schlechter abgeschnitten als die entsprechenden Modelle mit höherem C. Hinsichtlich des Hyperparameters γ ist die Modellgüte umso besser, je geringer γ ist. Der Vergleich der besten Modelle der jeweiligen Kernel hat vor allem gezeigt, dass die Modelle mit radialem Kernel den Modellen mit polynomialen Kernel überlegen sind. Dabei lag die Modellgüte zwischen den radialen und polynomialen Modellen nicht annähernd soweit auseinander, wie bei den entsprechenden Vergleichen zu den Modellen mit linearem Kernel. Diese Ergebnisse müssen allerdings mit Vorsicht genossen werden, da nicht der komplette Hyperparameterraum untersucht wurde, sondern nur einzelne, vorher festgelegte spezifische Werte. Der Einfluss der Hyperparameter kann hierbei noch wesentlich genauer untersucht werden, wie beispielsweise durch einen größer gewählten Hyperparameterraum oder auf Basis von mehr Datensätzen. Auch ließen sich die optimalen Hyperparameter mittels Tuning ermitteln und diese anschließend vergleichen. Des weiteren wäre es sehr interessant den Einfluss der Hyperparameter auf die Modellgüte in Abhängikeit von den datensatzspezifischen Eigenschaften zu untersuchen.

7 Literatur

[Ben-Hur and Weston, 2010] Ben-Hur, A. and Weston, J. (2010). *Data Mining Techniques for the Life Sciences*. Humana Press.

[Ferri et al., 2009] Ferri, C., Hernndez-Orallo, J., and Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Departament de Sistemes Informtics i Computaci, Universitat Politcnica de Valncia, Valncia 46022, Spain*.

[Gwava, 2016] Gwava (2016). *How Much Data Is Created On The Internet Each Day*. GWAVA. [https://www.gwava.com/blog/internet-data-created-daily - Letzter Aufruf: 14. Februar 2017].

[Hastie et al., 2016] Hastie, T., Tibshirani, R., and Friedmann, J. (2016). *Elements of statistical Learning - Data Mining Inference and Prediction*. Springer-Verlag New York.

[Hitachi, 2016] Hitachi, Q. M. (2016). *Connected cars will send 25 gigabytes of data to the cloud every hour*. QUARZ. [http://qz.com/344466/connected-cars-will-send-25-gigabytes-of-data-to-the-cloud-every-hour/ - Letzter Aufruf: 14. Februar 2017].

[Jüngling, 2013] Jüngling, T. (2013). *Datenvolumen verdoppelt sich alle zwei Jahre*. WeltN24 GmbH. [https://www.welt.de/wirtschaft/webwelt/article118099520/Datenvolumen-verdoppelt-sich-alle-zwei-Jahre.html - Letzter Aufruf: 14. Februar 2017].

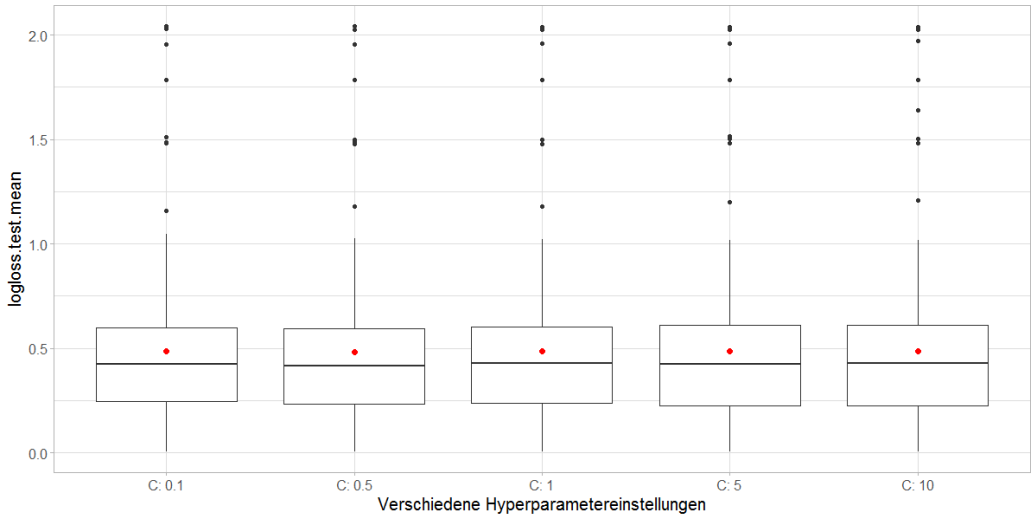
[Vanschoren et al., 2015] Vanschoren, J., van Rijn, J. N., and Bischl, B. (2015). Taking machine learning research online with openml. *Workshop and Conference Proceedings 41*.

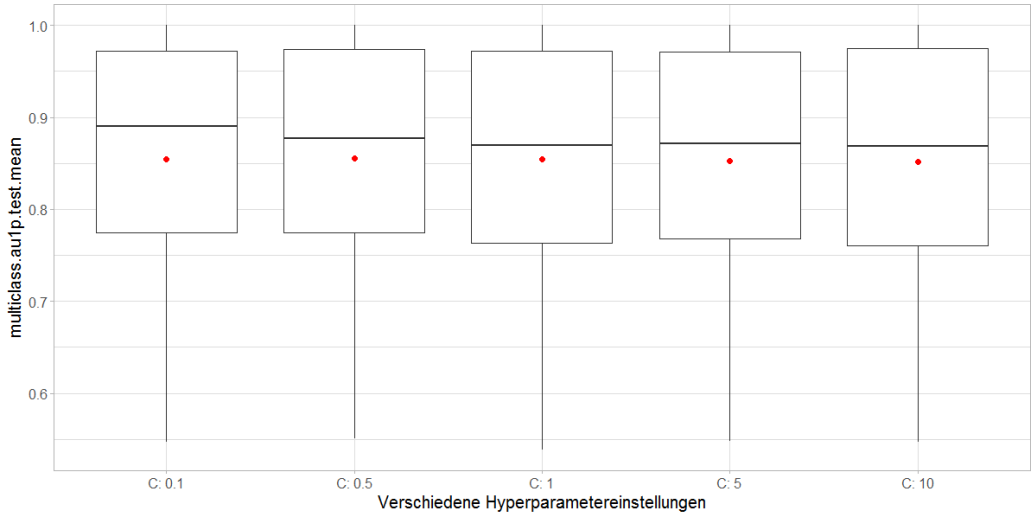
[Witten et al., 2013] Witten, D., James, G., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer-Verlag New York.

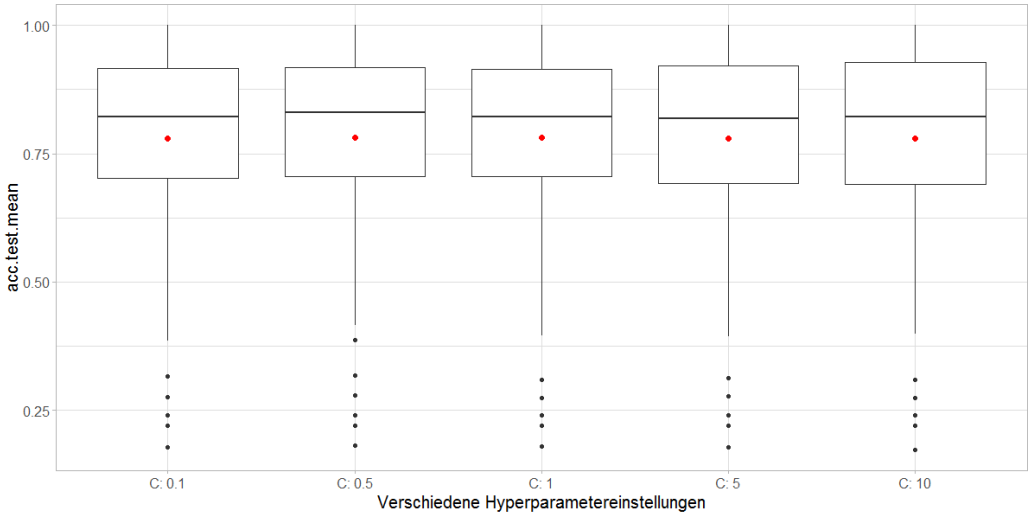
8 Anhang

Anhang für den linearen Kernel

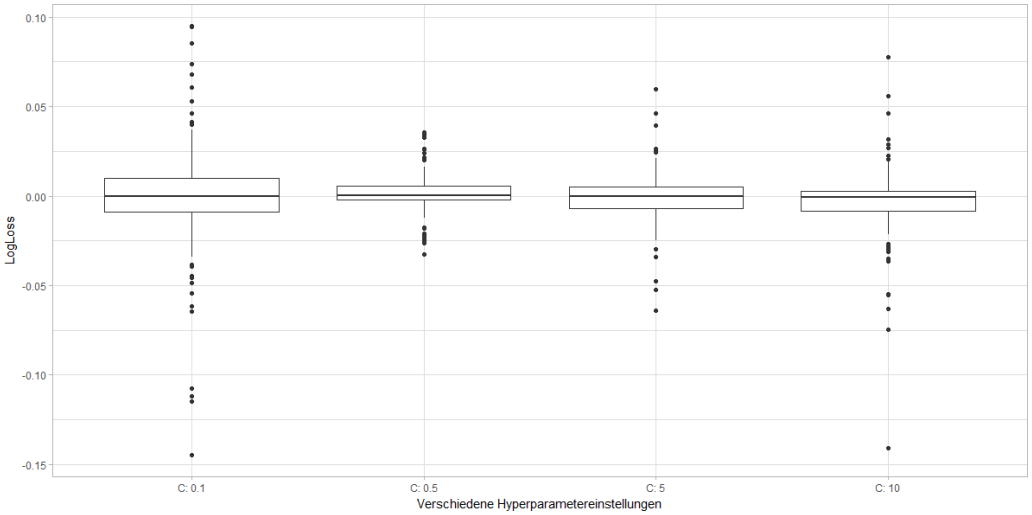
Boxplots der Verteilung, getrennt nach den Einstellungen für die Hyperparameter. Mit den restlichen Gütemaßen.

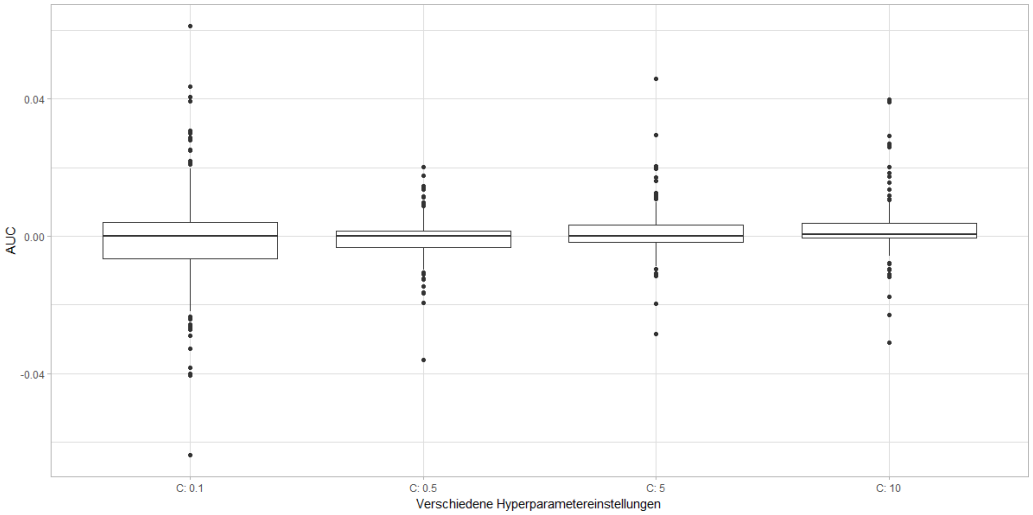


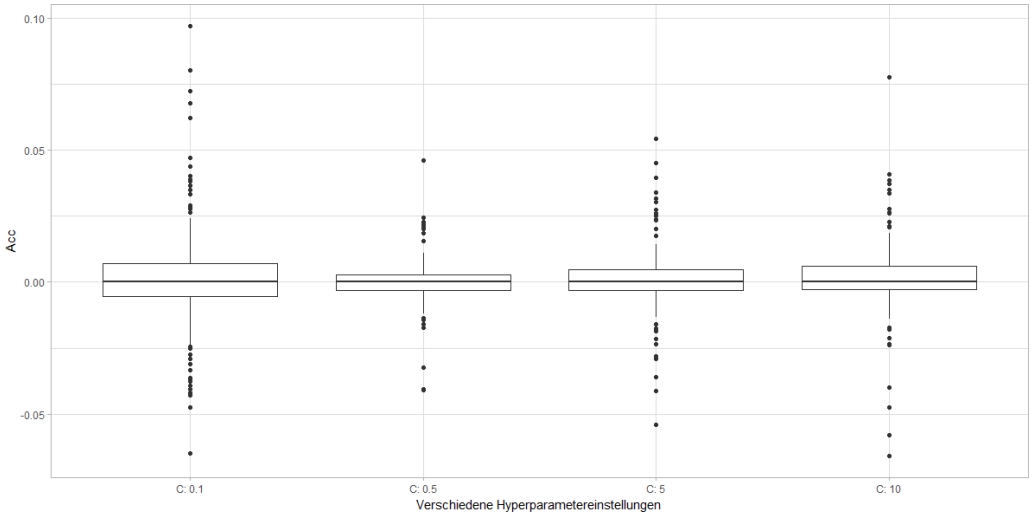




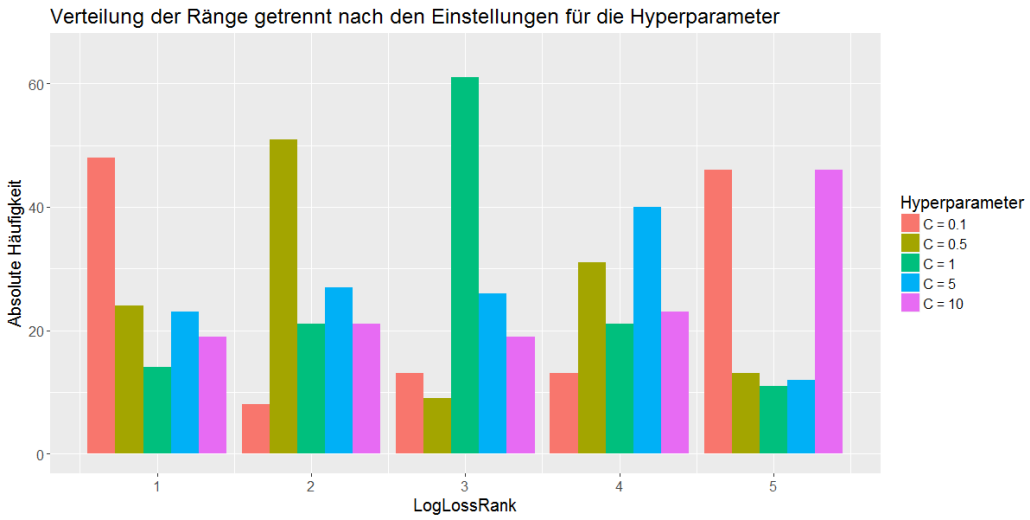
Boxplots der Differenzen zu dem Standardmodell für die verschiedenen Gütemaße.

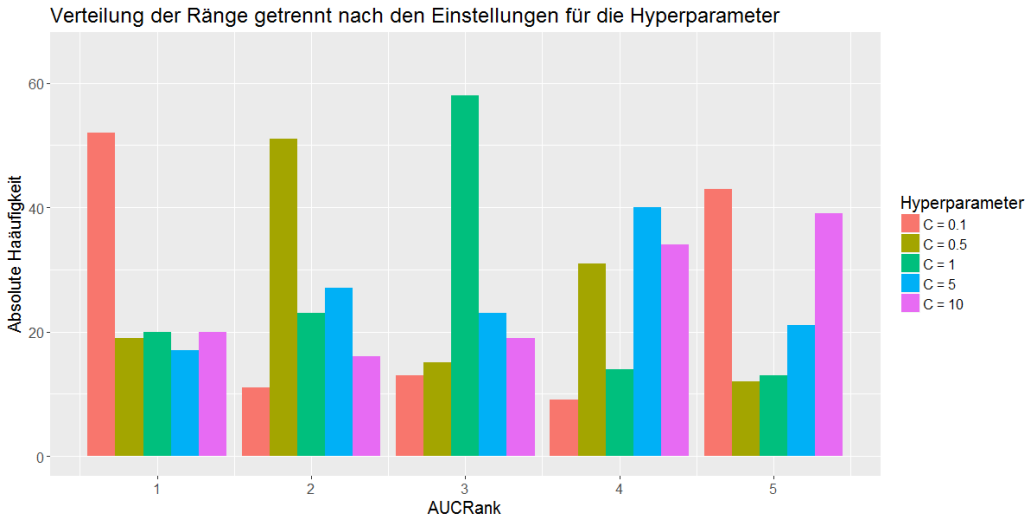


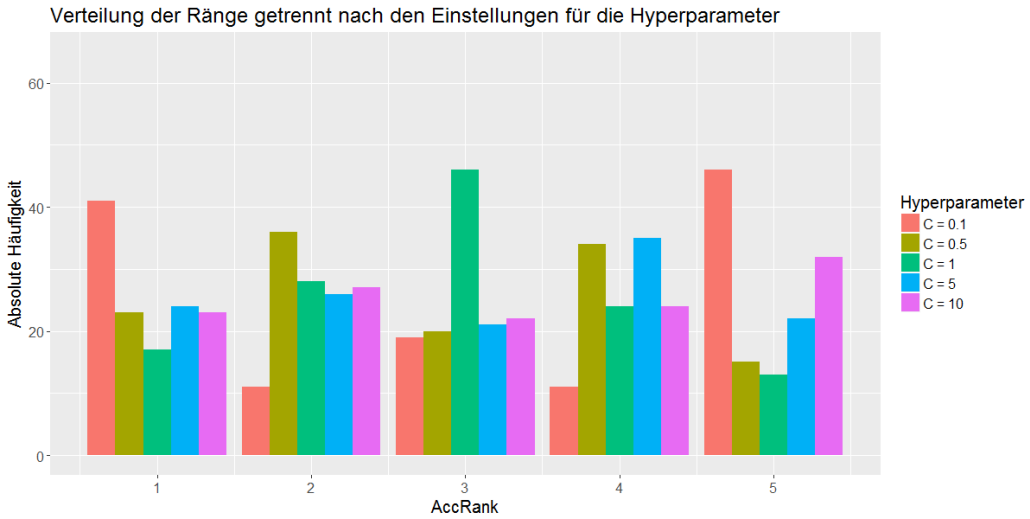

0.04
0.00
-0.04
AUC
C: 0.1
C: 0.5
C: 5
C: 10
Verschiedene Hyperparametereinstellungen


0.10
0.05
0.00
-0.05
Acc
C: 0.1
C: 0.5
C: 5
C: 10
Verschiedene Hyperparametereinstellungen

Ränge der Modelle für die restlichen Gütemaße

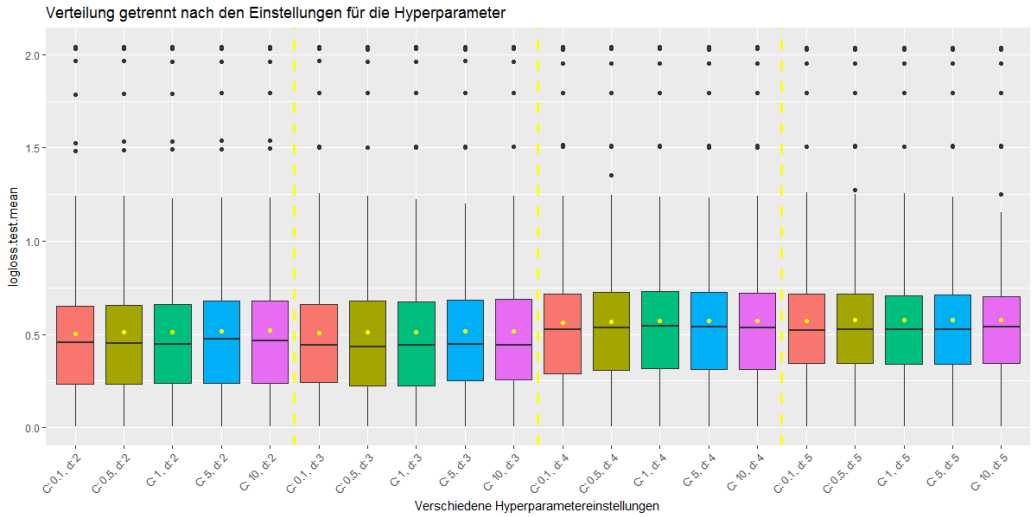


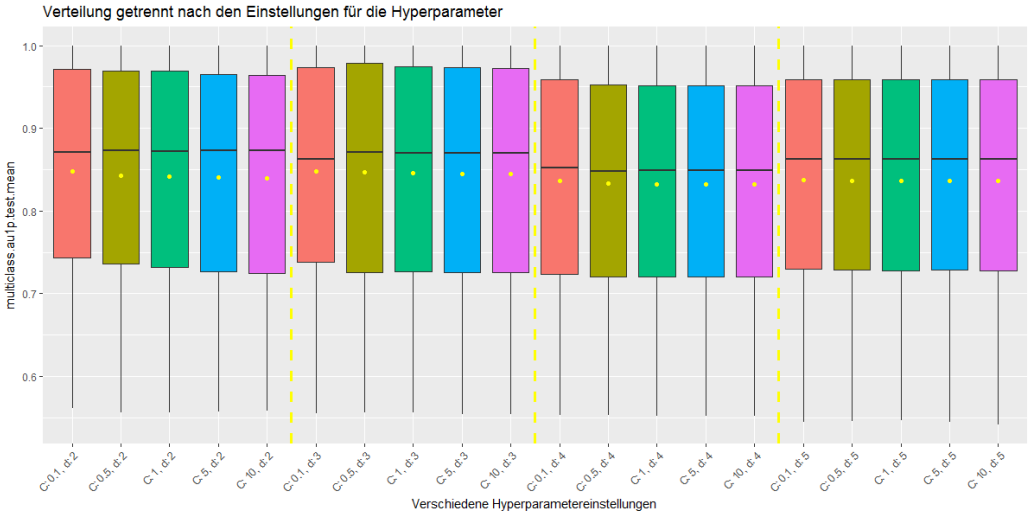


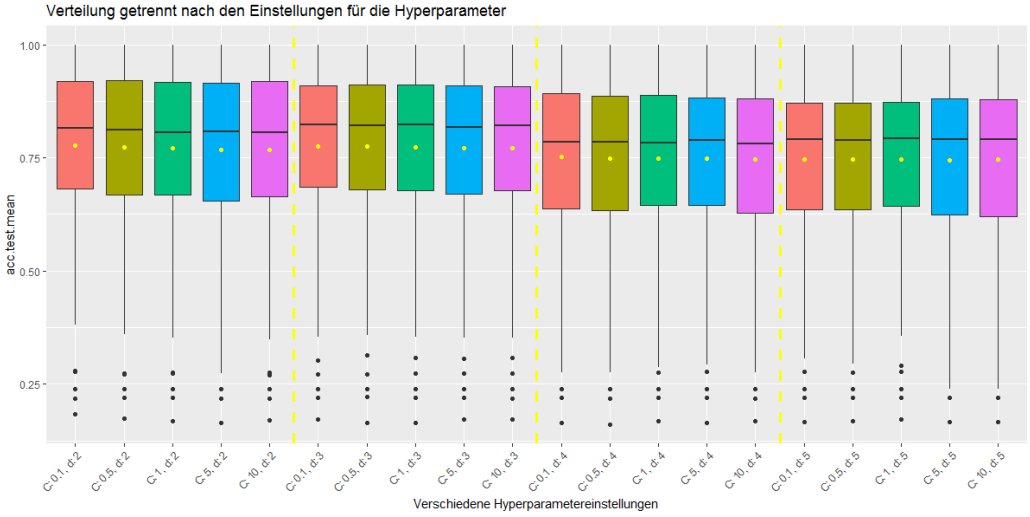


Anhang für den polynomialen Kernel

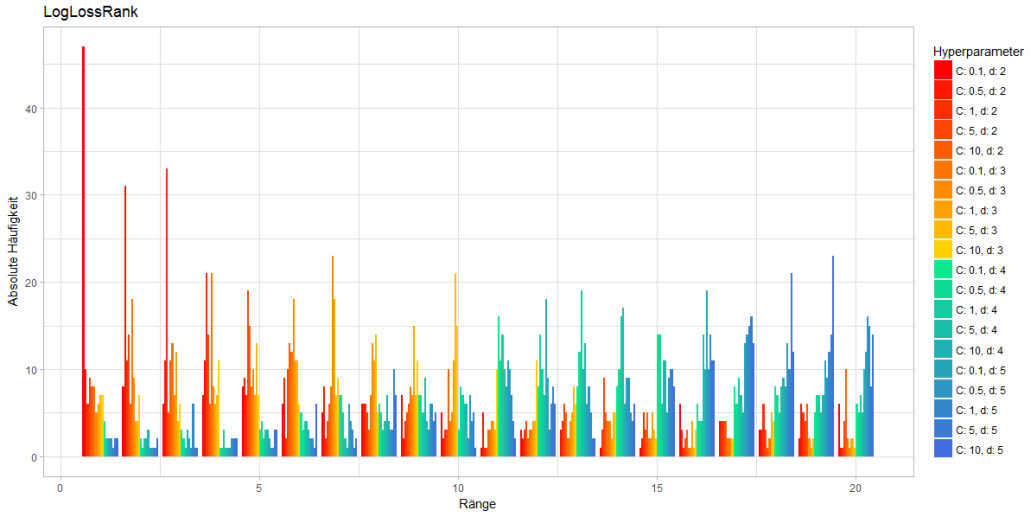
Boxplots der Verteilung getrennt nach den verschiedenen Hyperparametern für die anderen Gütemaße

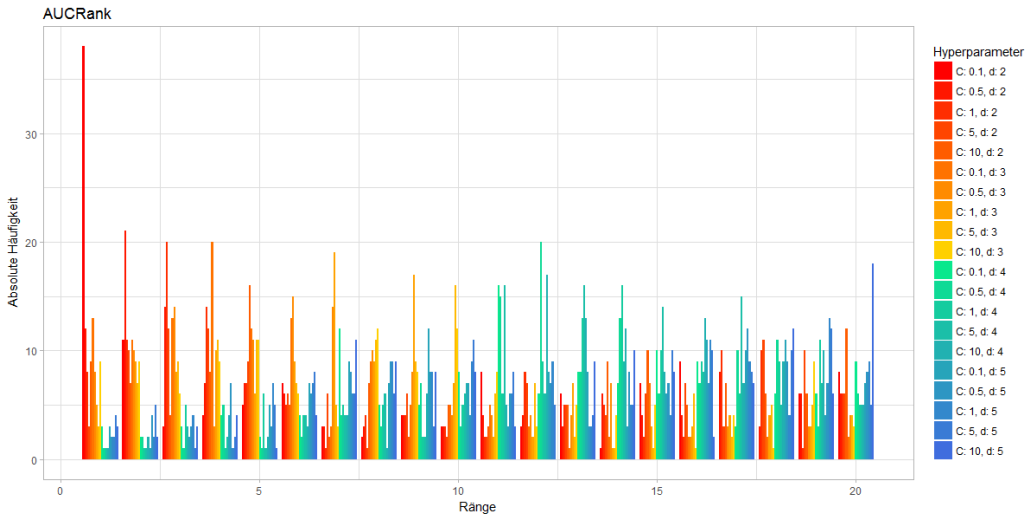


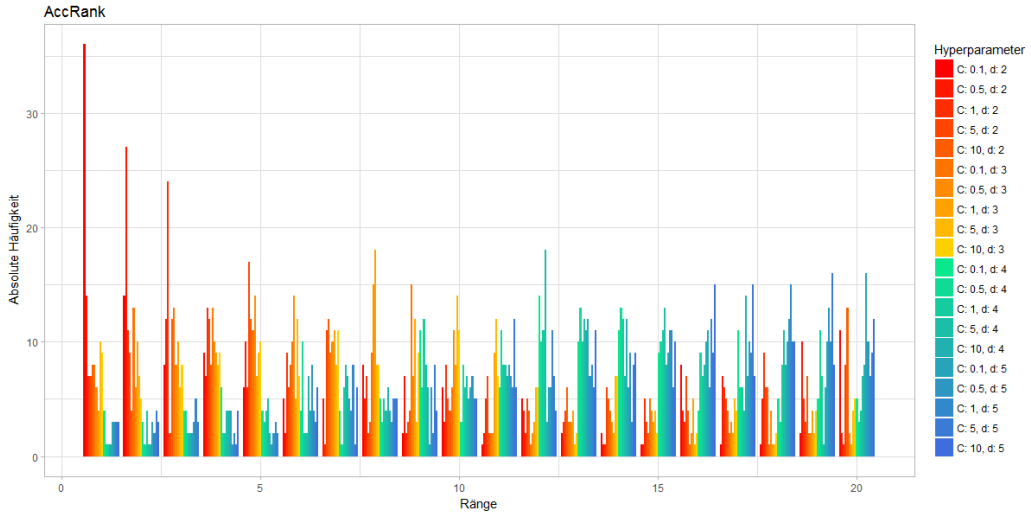

Verteilung getrennt nach den Einstellungen für die Hyperparameter
multiclass.au1p.test.mean
1.0
0.9
0.8
0.7
0.6
C: 0.1, d: 2
C: 0.5, d: 2
C: 1, d: 2
C: 5, d: 2
C: 10, d: 2
C: 0.1, d: 3
C: 0.5, d: 3
C: 1, d: 3
C: 5, d: 3
C: 10, d: 3
C: 0.1, d: 4
C: 0.5, d: 4
C: 1, d: 4
C: 5, d: 4
C: 10, d: 4
C: 0.1, d: 5
C: 0.5, d: 5
C: 1, d: 5
C: 5, d: 5
C: 10, d: 5
Verschiedene Hyperparametereinstellungen


Verteilung getrennt nach den Einstellungen für die Hyperparameter
acc.test.mean
1.00
0.75
0.50
0.25
C: 0.1, d: 2
C: 0.5, d: 2
C: 1, d: 2
C: 5, d: 2
C: 10, d: 2
C: 0.1, d: 3
C: 0.5, d: 3
C: 1, d: 3
C: 5, d: 3
C: 10, d: 3
C: 0.1, d: 4
C: 0.5, d: 4
C: 1, d: 4
C: 5, d: 4
C: 10, d: 4
C: 0.1, d: 5
C: 0.5, d: 5
C: 1, d: 5
C: 5, d: 5
C: 10, d: 5
Verschiedene Hyperparametereinstellungen

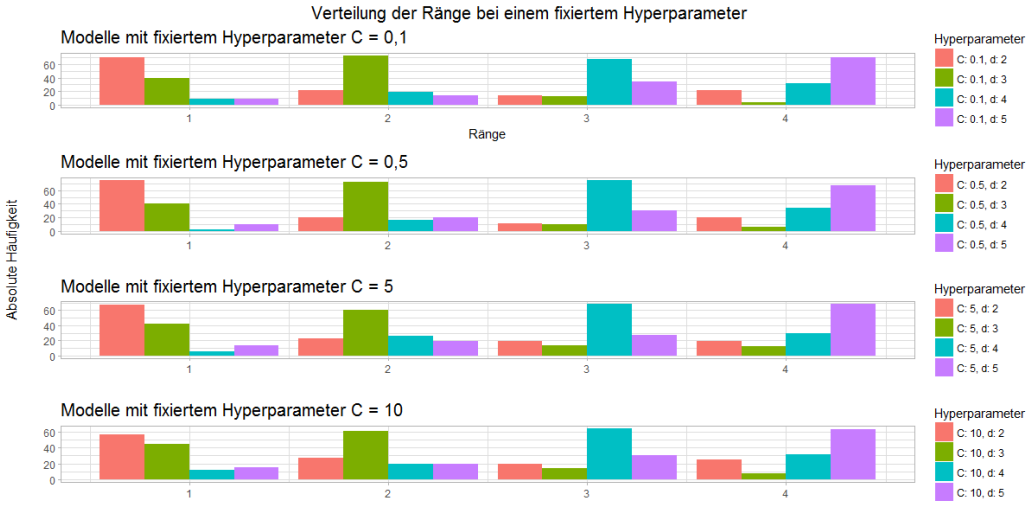
Ränge der Modelle anhand von anderen Gütemaßen

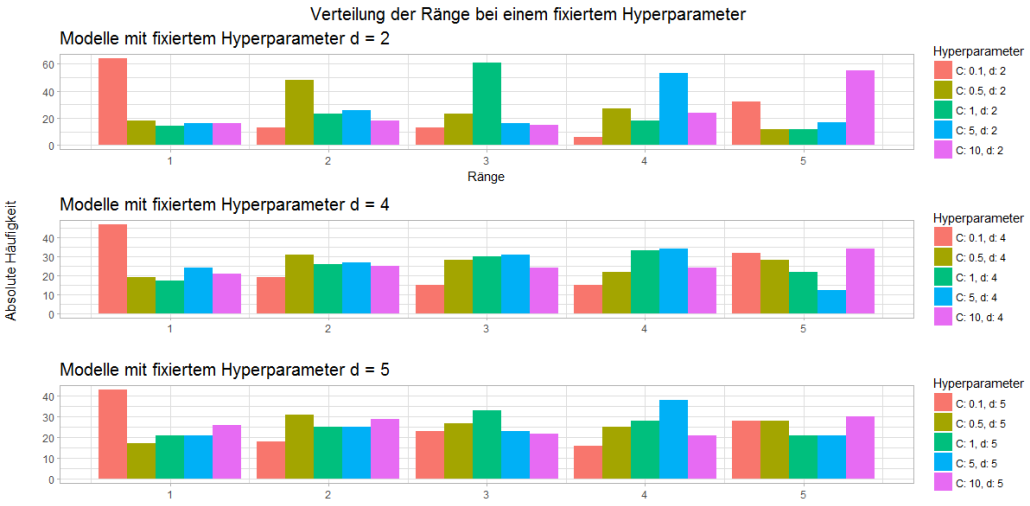






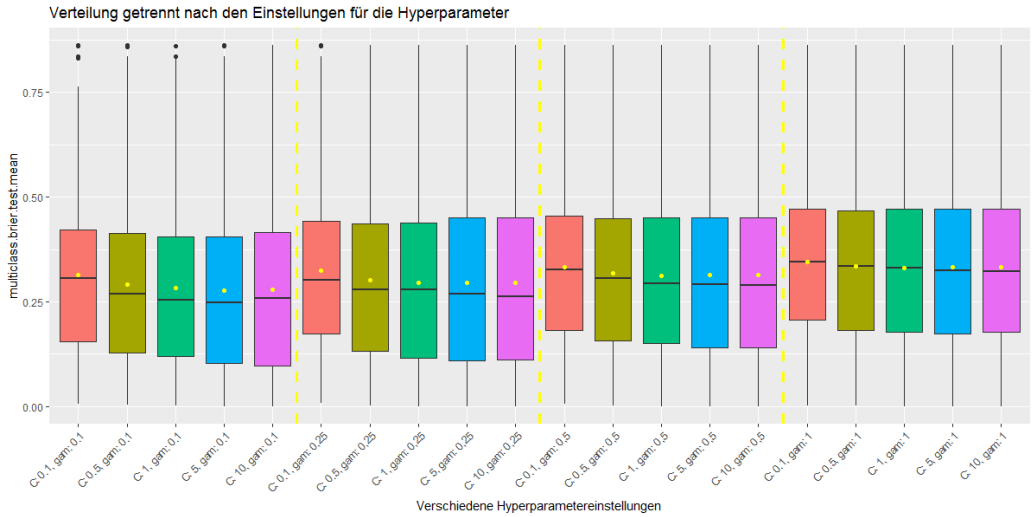
Ränge der Modelle bei andern fixierten Hyperparametern

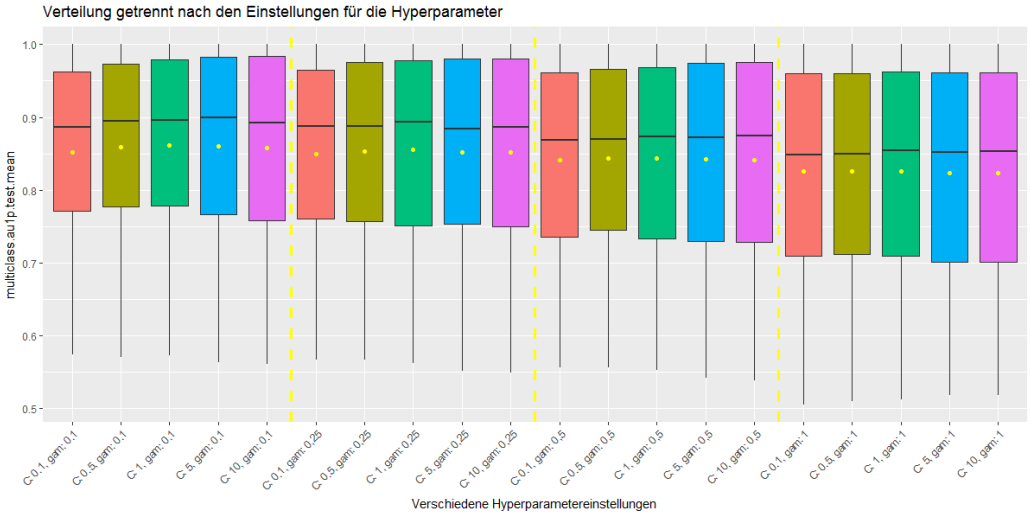


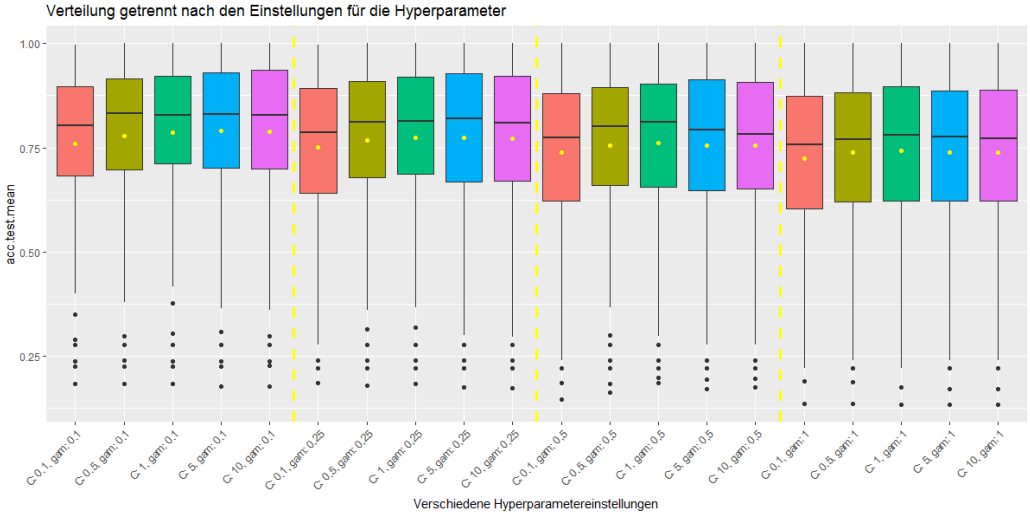


Anhang für den radialen Kernel

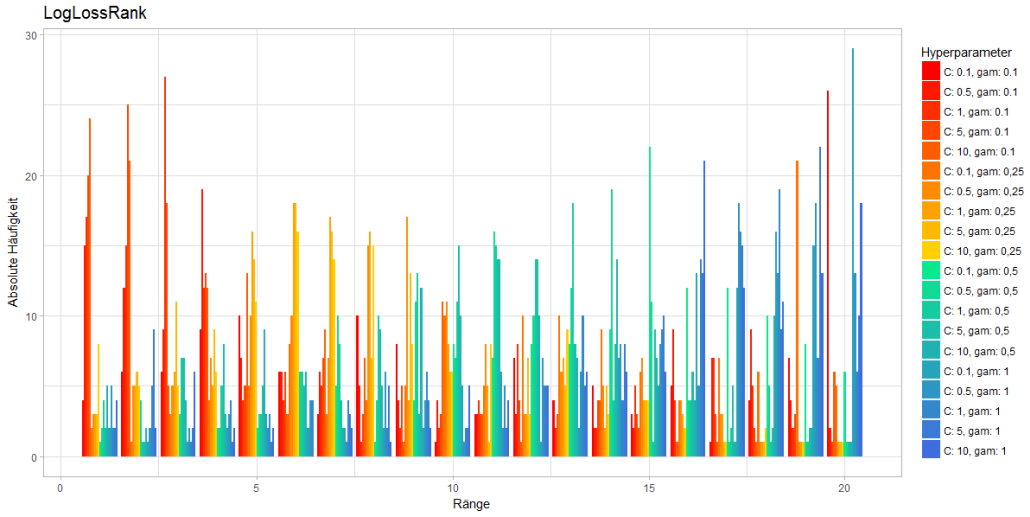
Boxplots der Verteilung getrennt nach den verschiedenen Hyperparametern für die anderen Gütemaße

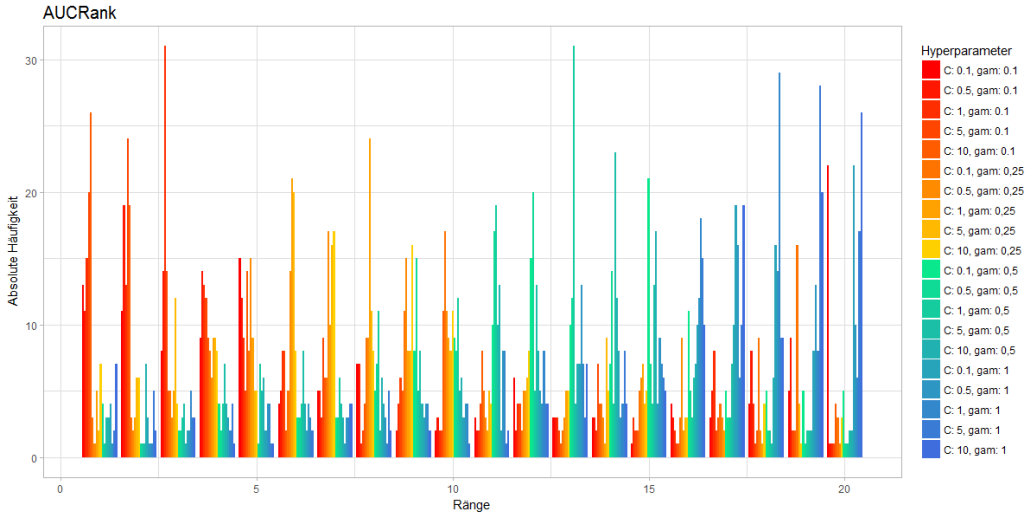


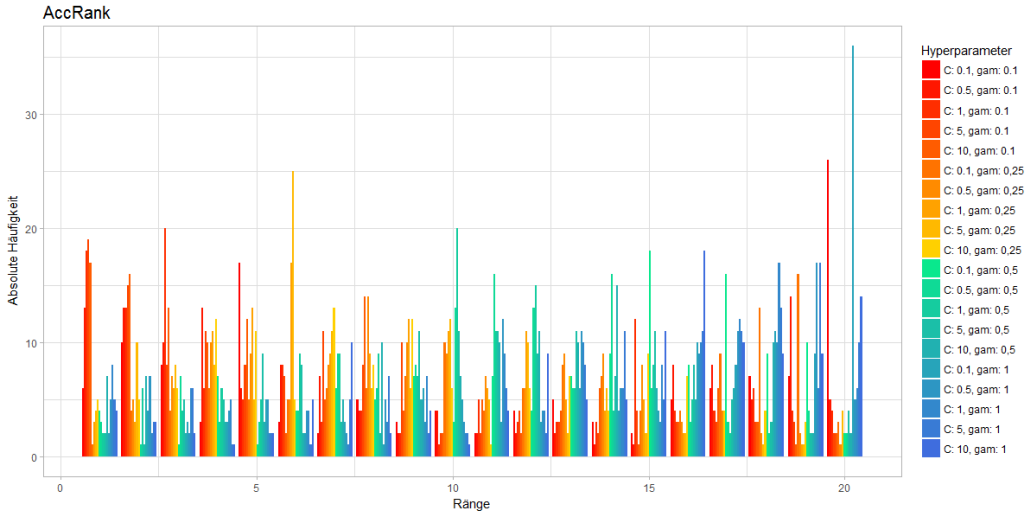

Verteilung getrennt nach den Einstellungen für die Hyperparameter
multiclass.au1p.test.mean
1.0
0.9
0.8
0.7
0.6
0.5
C: 0.1, gam: 0.1
C: 0.5, gam: 0.1
C: 1, gam: 0.1
C: 5, gam: 0.1
C: 10, gam: 0.1
C: 0.1, gam: 0.25
C: 0.5, gam: 0.25
C: 1, gam: 0.25
C: 5, gam: 0.25
C: 10, gam: 0.25
C: 0.1, gam: 0.5
C: 0.5, gam: 0.5
C: 1, gam: 0.5
C: 5, gam: 0.5
C: 10, gam: 0.5
C: 0.1, gam: 1
C: 0.5, gam: 1
C: 1, gam: 1
C: 5, gam: 1
C: 10, gam: 1
Verschiedene Hyperparametereinstellungen


Verteilung getrennt nach den Einstellungen für die Hyperparameter
acc.test.mean
1.00
0.75
0.50
0.25
C: 0.1, gam: 0.1
C: 0.5, gam: 0.1
C: 1, gam: 0.1
C: 5, gam: 0.1
C: 10, gam: 0.1
C: 0.1, gam: 0.25
C: 0.5, gam: 0.25
C: 1, gam: 0.25
C: 5, gam: 0.25
C: 10, gam: 0.25
C: 0.1, gam: 0.5
C: 0.5, gam: 0.5
C: 1, gam: 0.5
C: 5, gam: 0.5
C: 10, gam: 0.5
C: 0.1, gam: 1
C: 0.5, gam: 1
C: 1, gam: 1
C: 5, gam: 1
C: 10, gam: 1
Verschiedene Hyperparametereinstellungen

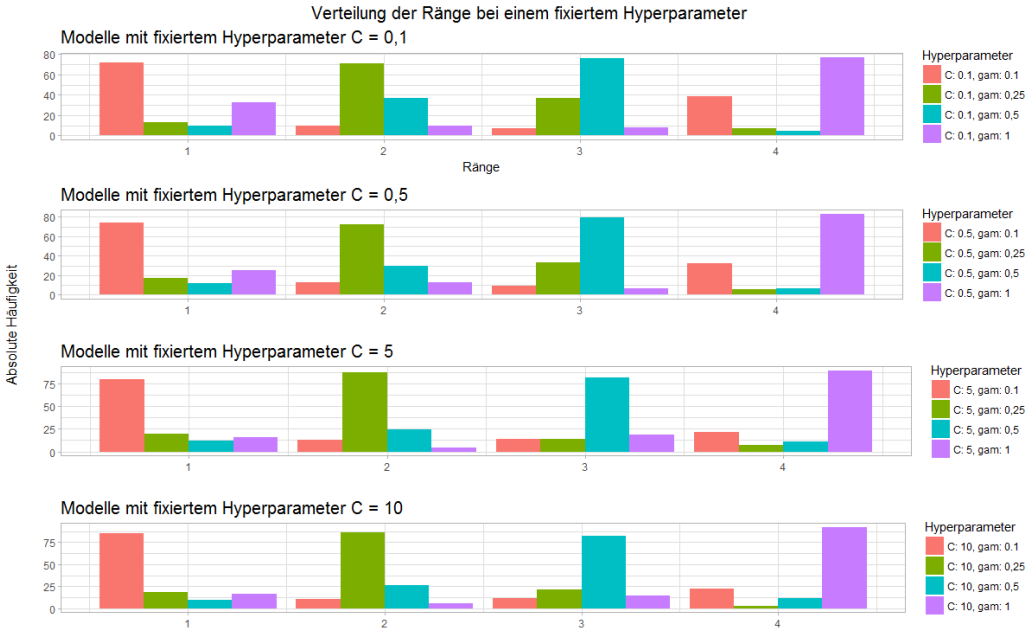
Ränge der Modelle anhand von anderen Gütemaßen

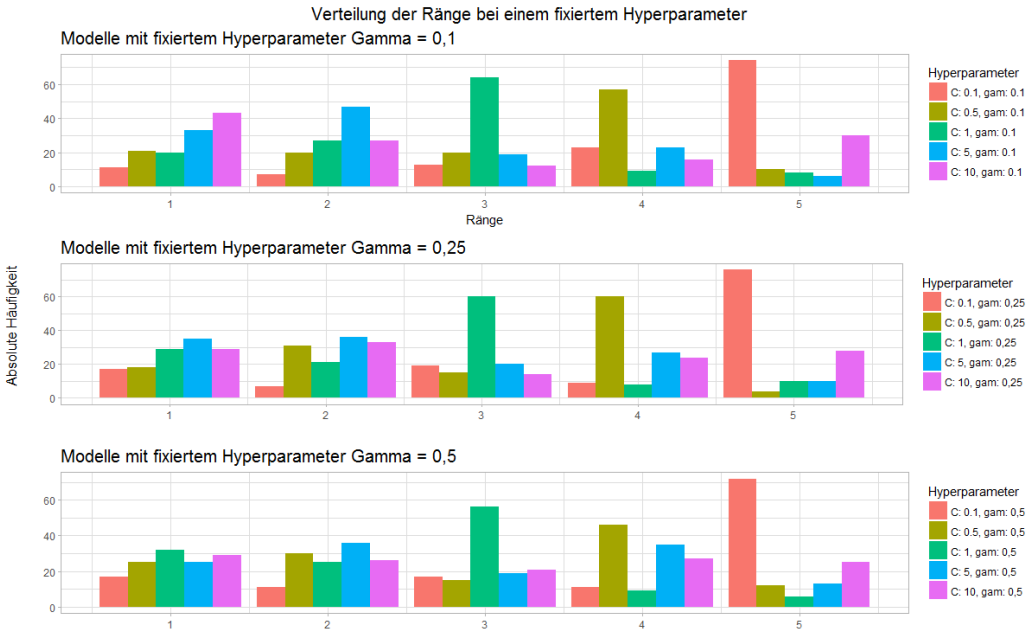






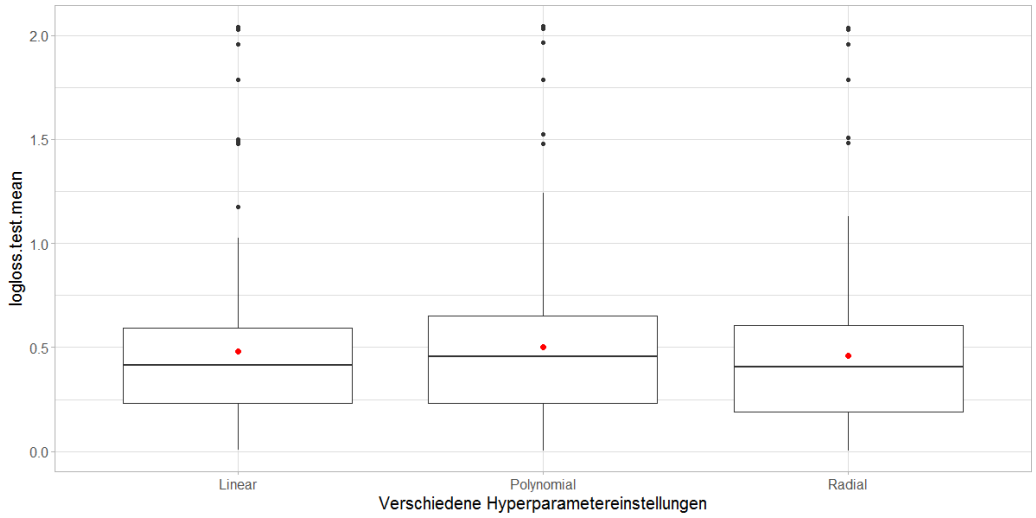
Ränge der Modelle bei andern fixierten Hyperparametern

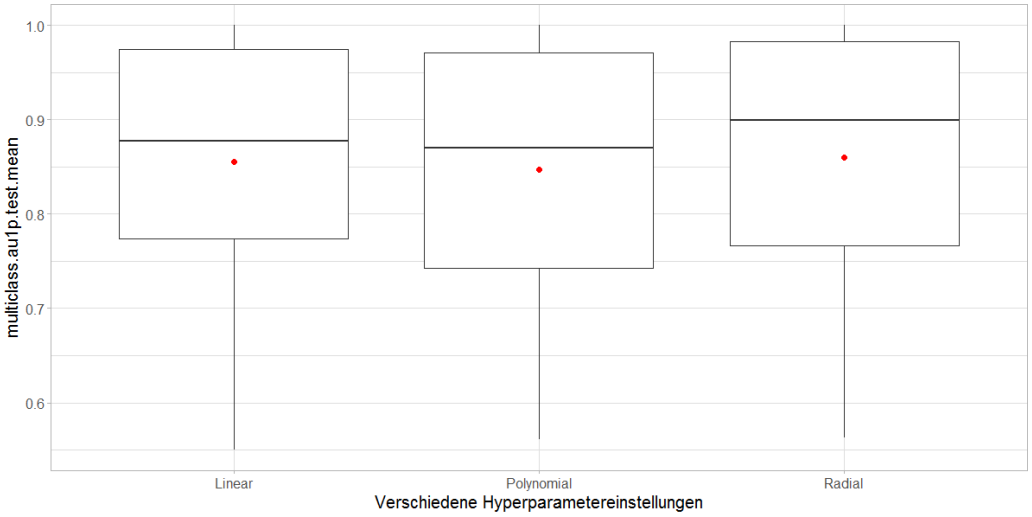


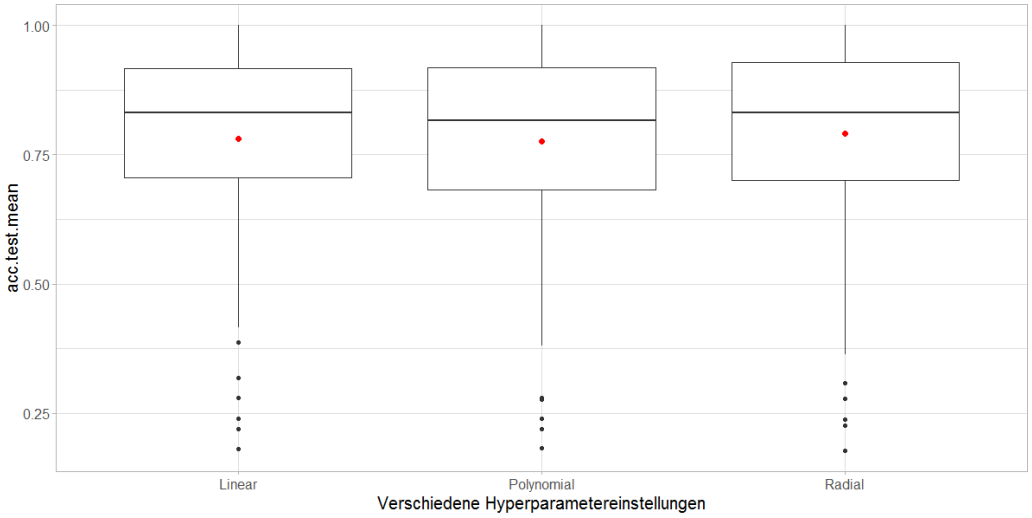


Anhang für den Vergleich der besten Kernel

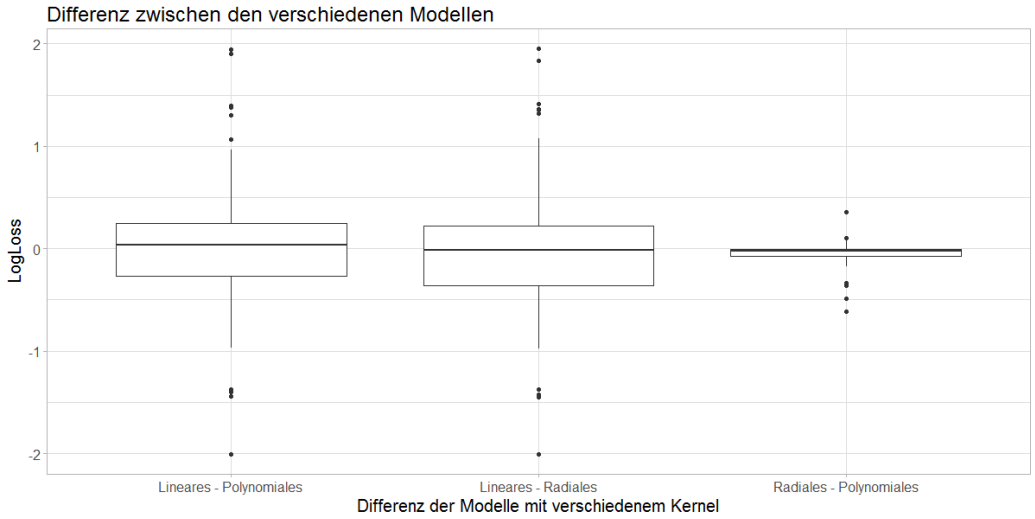
Boxplots der Verteilung getrennt nach den verschiedenen Kerneln für die anderen Gütemaße

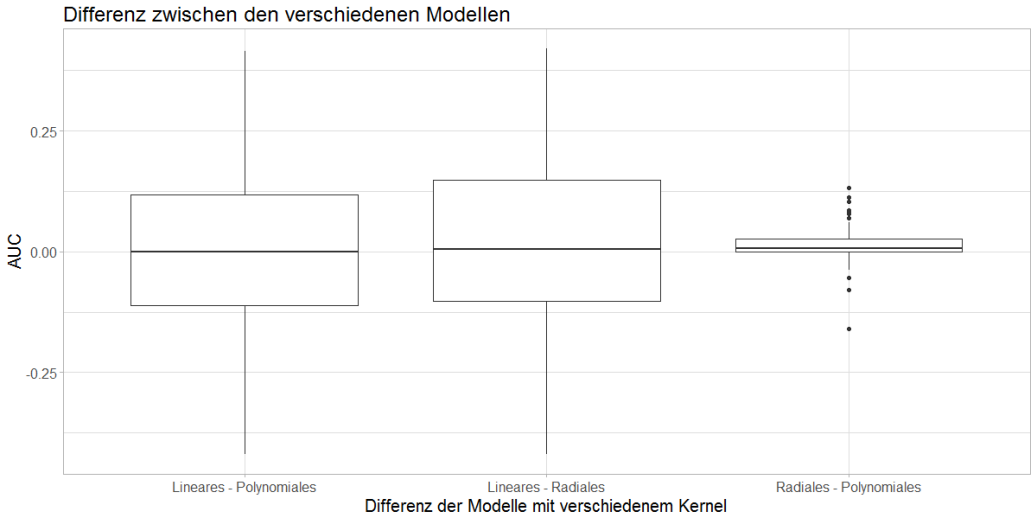


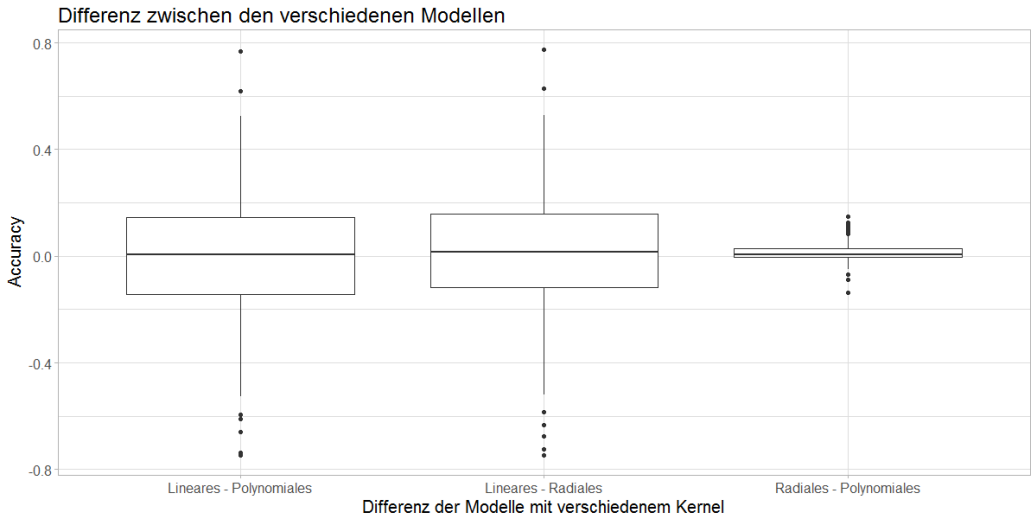

1.0
0.9
0.8
0.7
0.6
multiclass.au1p.test.mean
Linear
Polynomial
Radial
Verschiedene Hyperparametereinstellungen


1.00
0.75
0.50
0.25
acc.test.mean
Linear
Polynomial
Radial
Verschiedene Hyperparametereinstellungen

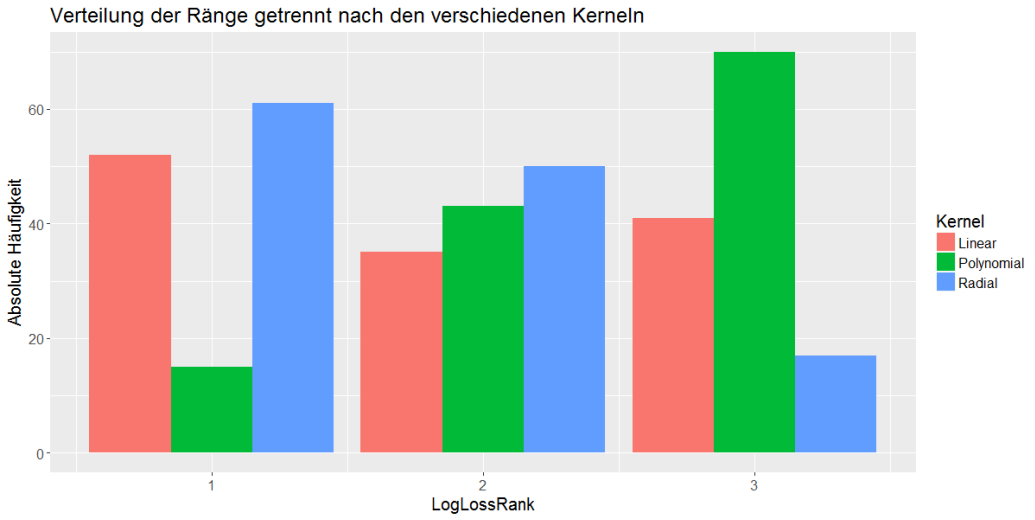
Boxplots der Differenzen zwischen den verschiedenen Modellen, getrennt nach den verschiedenen Kerneln für die anderen Gütemaße

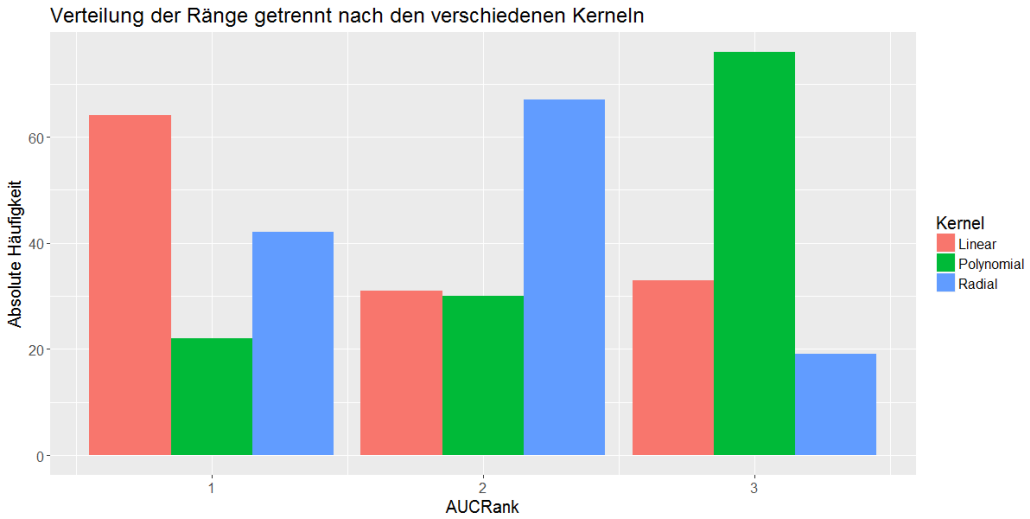


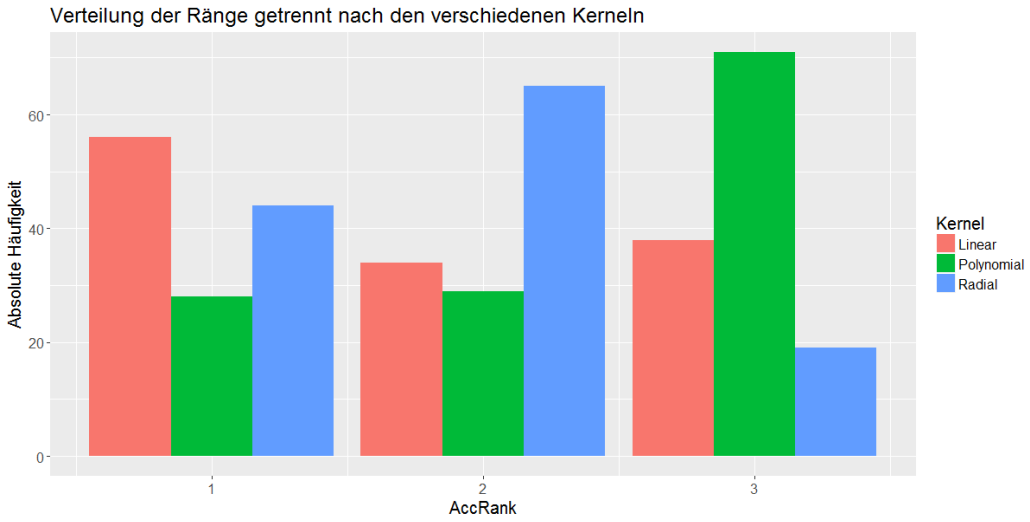




Ränge der verschiedenen Modellen, getrennt nach den verschiedenen Kerneln für die anderen Gütemaße




Verteilung der Ränge getrennt nach den verschiedenen Kerneln
Absolute Häufigkeit
0
20
40
60
1
2
3
AUCRank
Kernel
Linear
Polynomial
Radial


Verteilung der Ränge getrennt nach den verschiedenen Kerneln
Absolute Häufigkeit
0
20
40
60
1
2
3
AccRank
Kernel
Linear
Polynomial
Radial