

So, jetzt habe ich es geschafft. Langsam kommen alle an. Wir können nochmal an die Übung vom letzten Mal denken mit den Bankkunden. Was haben wir gemacht? Wir haben dann hier dieses Test & Score-Rückzug drangehängt und ich hatte euch gesagt,

Probiert mal vier verschiedene Algorithmen aus. Die hängen wir da jetzt nochmal dran. Also unsere absolute Baseline, Constant. Was hatten wir noch? Wir hatten logistische Regression, wenn ich mich recht erinnere. Und Naive Base. Und Tree haben wir natürlich auch immer dabei. Ich weiß gerade nicht mehr, wie tief der war und so. Aber spielt vielleicht auch nicht so eine Rolle. Mal schauen. Na ja, Tiefe 2 ist vielleicht ein bisschen zu wenig. Machen wir mal 5.

Und wir machen noch den Naive Bayes. Und dann hatten wir ja jede Menge Zahlen hier drin. Wir hatten, auch da spielt es nicht so eine Rolle, glaube ich, wahrscheinlich hatten die meisten von euch hier auf Random Something gestellt mit ein Drittel Testdaten. Und dann hatten wir all diese Zahlen. Erinnert ihr euch, was haben wir am Ende beschlossen? Also die Frage war natürlich, welcher Algorithmus? Okay, und warum?

Wir hatten ja so viele verschiedene Maße hier, die wir auch alle jetzt anfangen zu verstehen. Ja, danke.

Also wir hatten tatsächlich noch keine Kostenmatrix aufgestellt für dieses Szenario. Das machen wir heute. Aber wir hatten es sozusagen im Kopf. Also ich glaube...

Weil wir sie im Kopf hatten, haben wir uns hier auf eine der Metriken besonders gestürzt. Dennis, welche? Oder was war es da? Ich glaube, es ging um Reco. Und dort ist eben bei Neu-Bias 30% und bei Reco 70%. Und ich glaube, Precision ist eben die Wahl in der Hand, die aus Jahr-Positiv-Inputen, aus True-Inputen.

Und Recall ist dann, wie viele von diesen Waren wir dann auch tatsächlich finden. Und das bedeutet dann, wenn wir jetzt eben bei Position einen besseren Wert haben, dann klassifizieren wir zwar eigentlich eben 15% richtig, finden aber dann nur 23% und bei 9.5S wäre es dann ein unbegreiflicher Wert.

Ja, vielleicht, jetzt könnte es vielleicht verwirrend gewesen sein, für manche, für dich vielleicht auch, wenn wir es jetzt nochmal versuchen, da sitzt einer in dem Callcenter und fragt,

und ruft Leute an. Und das Modell hilft ihm zu entscheiden, wen er als nächstes anrufen soll. Wenn jetzt die Precision niedrig ist, was für eine Erfahrung macht dann dieser Mensch? Dass wir das nochmal so ausarbeiten, was das eigentlich praktisch bedeutet. Marc? Ja.

Ja, beides. Also wir hatten ja vielleicht auch nochmal gut, dass wir uns daran erinnern. Hier oben ist jetzt schon, weil sich Orange ja immer merkt, was ich beim letzten Mal gemacht habe, schon Yes eingestellt als Target sozusagen. Das heißt, diese Zahl, Station, betrifft sich darauf, dass es mir vorschlägt, jemanden anzurufen. Also weil ich Yes ausgewählt habe, geht es jetzt nicht darum, wovon es mir abrät.

Also die 30% würden jetzt bedeuten, wenn du beim ersten Teil deiner Geschichte bleibst, dass es häufig passiert. Also in 70% der Fälle ruft er jemanden an und da ist kein Interesse. Okay, das ist natürlich für den Menschen eine negative Erfahrung. Was passiert, wenn der Recall niedrig ist? Merkt es der Callcenter-Agent überhaupt? Vielleicht schon auch. Natürlich hängt beides zusammen.

Ja, Sterling, sag mal was. Also er merkt, wenn er viele Anrufe macht, kriegt er eher mehr Antworten, als wenn er wen dann nicht hat. Das ja, also absolut gesehen, klar, je mehr Anrufe, desto höher die Chance, dass auch noch einer dabei ist, der investiert.

Ja, also es ist gar nicht so einfach, das sozusagen spüren, dass der Recall niedrig ist. Es hängt natürlich zusammen. Also wenn er wenige Kunden überredet, dann bleibt der Recall niedrig. Und wahrscheinlich oder irgendwann merkt er vielleicht auch, dass das Modell ihm nichts mehr vorschlägt. Sozusagen wir haben es gesehen und wo haben wir es gesehen? Vielleicht machen wir das nochmal auf.

Wenn wir jetzt hier hingehen und unsere Confusion-Matrix dranhängen, dann sehen wir, ein Consent ist natürlich ganz schlimm, aber nehmen wir mal sowas hier, sowas eher pessimistisches. Von den 2800 Kunden in dem Testset schlägt uns der logistische

Regressierer überhaupt nur 101 vor, die wir anrufen sollen. Also wenn der jetzt nicht allein im Callcenter ist, dann vielleicht sind die zu viert, dann ruft jeder 25 an und dann hat das Modell keine Idee mehr. Das merken die dann einfach. Und wenn ich auf NaiveBase gehe, dann sehe ich, da werden insgesamt 600 Kunden vorgeschlagen. Dann haben die mehr zu tun. Also für die ist das eher schlecht. Aber sie merken natürlich auch, dass sie am Ende 180 Deals gemacht haben, während sie hier...

nur 59 gemacht hätten. Also haben wir nochmal Precision Recall wiederholt und jetzt hatten wir, also Dalia, du hattest von der Kostenmatrix gesprochen, wir haben sie im Kopf so ein bisschen, in dem Sinne, dass wir sagen, für die Bank zumindest, vielleicht nicht für den Agent, aber für die Bank ist es wichtig, dass wir diese Deals abschließen, weil wir annehmen, dass der Gewinn, den wir daraus ziehen, höher ist als die Kosten für die

einen Anruf, den wir umsonst machen. Also sozusagen Defaults Positives sind billiger als Defaults Negatives. Also die Opportunitätskosten, die entstehen, wenn wir jemanden nicht anrufen, der interessiert gewesen wäre. Okay, die Erinnerung ist wieder da, oder? Jetzt hatte ich noch eine Folie nicht mehr geschafft beim letzten Mal und sie wäre noch wichtig gewesen für zwei der Quizfragen, aber vielleicht habt ihr es auch so rausgekriegt. Ich denke mal, ich wollte noch diese Folie gezeigt haben.

Da geht es so ein bisschen darum, dass man sich einerseits nicht nervös machen lässt von Anforderungen, die auf einen einstürmen und andererseits, dass man es schafft. Also es geht immer darum, dass man oder dass ihr vielleicht irgendwann in der Situation seid, dass ihr solche Modelle entwickelt und dann mit Business Leuten, mit Stakeholdern darüber kommunizieren müsst. Also das ist jetzt meine persönliche Präferenz, was ich

ja schon gesagt habe, dass ich Fan bin von kostenbasierter Evaluation und ich glaube, dass es sich einfach für Kommunikation sehr gut eignet, wenn ich zwei Modelle habe und sagen kann, mit dem Modell machen wir so viel Gewinn, mit dem Modell machen wir so viel Gewinn, das kann ja jeder verstehen, was das bedeutet und warum es Sinn macht, dann das Modell zu nehmen, bei dem wir mehr Gewinn machen. Das muss man nicht viel erklären. Wenn man mit Precision und Recall kommt, erstmal muss man erklären, was das ist und dann

Sagen die Leute vielleicht, ja, was ist denn jetzt wichtiger? Oder sie schießen sich auf eins von beiden ein, weil sie es besser verstanden haben oder was auch immer. Wenn ihr versucht, ihnen Area under the Curve Wert zu erklären, dann wird es schon ganz schwierig. Das ist einfach schwierig zu

kommunizieren, meiner Erfahrung nach. Es ist nicht immer möglich, eine Kostenmatrix aufzustellen. Also wir haben ja beim Aufstellen von Kostenmatrizen, erinnert sich noch jemand, was für eine wir aufgestellt haben letztes Mal? Also wir haben ja zwei aufgestellt, aber...

Wenn ihr euch noch an eine erinnert, bin ich schon happy. Worum ging es? Fälle von Versicherungsbedruck. Ja. Genau. Jetzt muss ich mich nur erinnern, worauf ich hinaus wollte. Genau. Ja, also was wir da gemacht haben, war, wir haben zum Beispiel angenommen, was ist der durchschnittliche Kreditbetrag? Ja, also wenn das...

Sagen wir mal, wenn wir uns fokussieren auf eine bestimmte Art von Kredit, vielleicht für Autos oder so, dann ist es vielleicht plausibel, dass man in einem gewissen Range bleibt, was Autos eben so kosten und dann auch annehmen kann, wie viel Geld man da verliert, wenn jemand das nicht zurückzahlt. Wenn wir gar nicht wissen, wofür das Geld benutzt werden soll, dann haben wir eine Riesenstreuung von möglichen Kreditsummen,

Und dann funktioniert die Kostenmatrix vielleicht. Also wir sehen dann, dass irgendwie bestimmte Kosten sich ergeben für unser Modell, aber ob das in der Praxis dann so stimmt, ist vielleicht zweifelhaft. Also das ist das, was hier gemeint ist mit Kosten gut quantifizieren. Also gut auch in dem Sinne von einigermaßen sicher, ohne dass die Streuung zu groß ist oder ich wilde Annahmen machen muss, die ich nicht verifizieren kann.

Aber wenn man es kann, dann bin ich eben großer Fan. Und das ist jetzt das, wo ich gesagt habe, lasst euch nicht irre machen sozusagen. Vielleicht gibt es Forderungen, die an euch herangetragen werden. In der einen Quizaufgabe hatte ich es auch so dargestellt, dass ihr vielleicht Konkurrenz bekommt von anderen Leuten, die auch ein Modell entwickelt haben. Und die wählen dann natürlich die Metrik, die sie am besten darstellen, stehen lässt.

Und dann, also in dem Beispiel kommt dann der Chef und sagt, ja, die sind aber besser. Und die Metrik finde ich cool, die habe ich verstanden. Und die will ich auch optimiert haben. Und die Frage ist dann immer so ein bisschen, kann man selber wirklich rational drangehen und sich überlegen, was ist die beste oder die wichtigste Metrik? Und bin ich nicht vielleicht anhand dieser Metrik doch besser als die Konkurrenten?

Und insbesondere das, was hier steht, ist auch, dass man nur eine Metrik haben sollte, die man optimiert. Also wenn zum Beispiel jemand sagt, wir wollen das Modell in Echtzeitszenarien einsetzen und für jede Entscheidung hat dein Modell drei Sekunden maximal, was dann schon zum Beispiel den K-Nearest Neighbor in vielen Fällen rauskickt, dann...

muss man nicht das Modell finden, was am schnellsten entscheidet. Man muss nur eins finden, was schneller als drei Sekunden entscheidet. Und dann kann man unter all denen, die das schaffen, das auswählen, was am genauesten vorher sei. Ich würde diese zwei Quizfragen kurz anschauen, aber wir können natürlich auch andere Fragen anschauen, zu denen ihr noch Zweifel habt. Ja.

Die Evaluation ist früher. Das ist das Problem. Das ist natürlich dramatisch. Ich glaube dir auch. Das mache ich dann in der Pause. Ich mache es in der Pause. Zurück zur Frage, was

Wollt ihr vielleicht noch anschauen? Jetzt außer Frage 5 und 6, die zu dieser Folie gehörten, die ich nicht nur nicht besprochen habe beim letzten Mal, sondern noch nicht mal für euch veröffentlicht habe. Bei der ersten Frau. Ja, die Lernkurve, okay. Wieso? Warum bei 200? Also was man sich immer...

überlegen muss bei so einer Lernkurve ist, dass du die nie im Ganzen sehen wirst. Also die Überlegung, die hier stattgefunden hat bei mir, ist, dass man dies von links so entwickelt. Also man fängt an, Beispiele zu labeln. Und man will ja immer wissen, muss ich noch weicher machen? Wird es sich noch weiter verbessern? Und da ist eben der Klick. Also ich sehe hier bei 150 habe ich 90% Accuracy.

Gut, da gab es noch eine Verbesserung gegenüber 100. Also es geht sowieso um die grüne Linie. Die sehe ich, wenn ich von links anfangen, immer oben. Das heißt, selbst wenn die sich hier am Ende treffen, werde ich mich, also da komme ich ja nie an. Ja gut, jetzt bin ich hier. Dann sage ich, na gut, wir hatten beim letzten Mal noch ordentlich Verbesserung. Vielleicht kriege ich noch 95 hin. Nochmal label ich nochmal 50. Und dann, ich bin ja auch, also hier steht,

dass es ein großer Aufwand ist. Ich bin ja auch faul und überhaupt geht ja auch um Geld. Wirtschaftlich sinnvoll würde ich dann sagen, jetzt geht es runter, jetzt höre ich aber auf. Weil ich nicht mehr damit rechne. Also das ist natürlich schade. Ihr hättet doch noch was gewinnen können, aber ich würde wahrscheinlich... Aber man merkt ja erst, dass die Höhlenkurve flach ist bei etwa 352. Also ich kann natürlich nicht in die Leute reinschauen. Es könnte auch sein, dass jemand vielleicht doch sagt,

okay, jetzt ist es runtergegangen, aber ich glaube dran, wenn ich nochmal 50 label, wird es besser. Aber sozusagen der Gewinn ist ja auch nicht so wahnsinnig groß, wenn man sich überlegt. Wenn der Aufwand fürs Labeln wirklich hoch ist, ist es tatsächlich auch schon fraglich, ob es sich wirklich lohnen würde. Und in der Praxis würde ich vermuten, dass die meisten Leute da einfach aufhören, weil sie sagen,

Ich zue ich mir nicht an, nochmal 50 zu labeln, wo es doch jetzt gerade schlechter geworden ist. Deswegen war das so die Überlegung. Also die Überlegung, zu der ich euch auch zwingen wollte, dass man nochmal sich erinnert, dass man eigentlich nicht die gesamte Kurve sehen wird. Jetzt habe ich dich so halb überzeugt. Gab es noch eine andere Frage, die ihr diskutieren wolltet? Ja.

Ja, also die Accuracy sozusagen, die ist wirklich leicht zu berechnen. Also das kannst du bei so Confusion-Maten einfach, insbesondere wenn ich dir sage, wie groß die Testmenge ist, dann musst du sie gar nicht erst alle aufsummieren.

Und hier ist es sogar noch einfacher, weil ich die Zahl so schön rund gewählt habe. Das heißt, du musst wirklich immer gucken, wie viele hat er richtig gemacht. Die findest du auf der Diagonalen. Also 68 hat er richtig gemacht. 8 plus 60. Und da es 100 waren, ist es dann direkt auch die Zahl, die du hier einpippst. Und hier eben 93. Also jetzt können wir die Aufgabe aber auch nutzen. Ist klar. Wir können die Aufgabe auch nutzen, um uns eben nochmal über kostenbasierte Evaluationen

klar zu werden. Das war wieder unser Wintercheck. Auch da haben wir ja den Fall, dass Recall wichtiger ist als Precision, in dem Sinne, dass der Gewinn, den wir haben, wenn jemand sein Fahrrad bringt, sehr viel höher ist als die Kosten, wenn wir den Brief umsonst geschickt haben. Also der falsche Alarm sozusagen. Und man sieht ja jetzt hier, dass A das mutigere Modell ist. Man sagt 38 Mal ja und B ist mehr so konservativ, sagt nur 3 Mal ja.

Und eigentlich wollen wir das hier lieber. Aber man sieht natürlich, dass die Genauigkeit von A oder die Accuracy niedriger ist, weil es eben so viele false positives produziert. Wir wollen aber vor allem belohnen mit unserer Kostenmatrix, wenn hier oben auch wirklich Interessenten gefunden werden. Und jetzt muss man natürlich das Berechnen einfach durch multiplizieren. Ihr erinnert euch.

Und dann sieht man eben, dass A die niedrigeren Kosten hat oder sagen wir die höher negativen, also den höheren Gewinn. Ich rechne es jetzt einfach mit umgekehrtem Vorzeichen, also 8 mal 39 muss ich rechnen, minus 30. Das heißt, die Kosten muss ich jetzt natürlich mit negativen Vorzeichen eingeben, weil ich es falschrum jetzt gemacht habe, hat diese Matrix den Gewinn von 282 Franken. Also

Acht Leute kommen mit ihrem Fahrrad, jeder von denen bringt mir diesen Gewinn von minus 39, also 8 mal minus 39 plus 30. Das hier, hoffentlich habe ich es richtig gemacht. Und bei dem anderen, da ist es einfacher noch, da ist es einfach minus 3 mal 39, 3 mal minus 39, also ein Gewinn von 117, also nicht mal halb so viel Gewinn. Deswegen würde ich dann...

annehmen, weil mir die Genauigkeit nicht so wichtig ist wie die Kosten. Warum sieht das bei der E3 mal Ja? Also, der E sagt 3 mal Ja und das stimmt. Das heißt, wir haben 3, die ihr Fahrrad bringen und hier unten keine Vs-Positives. Also keine Briefe, die wir umsonst schicken. Alles klar?

Gut, zwei haben wir übersprungen, das war okay für euch. Das war vier nochmal. Ja, das ist eine gute Frage. Also ich sollte vielleicht einfach immer die englischen Begriffe verwenden.

Accuracy und Precision, ich weiß nicht, wenn man das irgendwie ins Wörterbuch eingibt, dann kommt vielleicht beide Male Genauigkeit raus. Hier meine ich die Accuracy, ja. Vielleicht war das auch das, Marc, was dich irritiert hat. Du hast die Precision ausgerechnet. Ja, okay. Ich nehme mir mal vor, dass ich einfach nur noch die englischen Begriffe verwende, weil die sind irgendwie klarer.

Frage 4 war klar, oder? Dann gibt es auch eine schöne Erklärung, glaube ich. Hier gab es ja auch Erklärungen, aber jetzt können wir es sozusagen nochmal in Relation zu der Folie setzen, die ich euch gerade gezeigt hatte. Das erste ist eben dieses Szenario, wo ihr mehrere Forderungen erfüllen müsst, sozusagen. Also möglichst genau und hier steht jetzt extra nicht, möglichst kleines Modell und möglichst schnell erfüllt.

Sondern es steht nur da, es darf nicht größer sein als 15 Megabyte und es darf nicht länger als 10 Sekunden brauchen. Und dann ist es eben genau so, jetzt lasst euch nicht stressen. Oder wenn jetzt jemand zu euch sagt, es muss möglichst klein, möglichst schnell und möglichst genau sein. Also möglichst hohe Accuracy oder was auch immer eure Metrik ist. Vielleicht AOC soll möglichst hoch sein.

Gebt euch damit nicht zufrieden. Sagt, so kann ich nicht arbeiten. Sagt mir, gibt es eine Größe oder eine Zeit, bis zu der es akzeptabel ist. Meistens gibt es das. Also können die euch dann sagen, ja, es darf bis zu zehn Sekunden dauern. Und alles, was darunter ist, ist schon okay. Und wenn ihr das habt, dann könnt ihr sagen, okay.

Alle Modelle, die wir lernen, müssen das erfüllen. Das heißt, alle, die länger brauchen, fliegen sofort raus. Und alle, die mehr Platz brauchen. Also hier braucht ja keines mehr Platz als gefordert. Es gibt nur eins, was länger dauert. Länger braucht. Genau, das fliegt also schon mal raus. Das heißt, das hätte ich jetzt nicht machen sollen. Und alle anderen sind noch im Spiel. Und da werde ich jetzt einfach das, was die höchste Genauigkeit hat. Also Accuracy meine ich hier wieder. Und

Jetzt könnt ihr schauen, das ist weder das schnellste noch das kleinste. Aber es erfüllt beide Bedingungen und es ist das genaueste, also es ist noch nicht mal das genaueste unter allem. Das heißt, wir haben wirklich keine dieser Metriken global optimiert. Aber wir haben alle Anforderungen erfüllt. Verwirrend oder klar? Also letztlich, wenn ihr so Anforderungen bekommt, die mehrere Metriken irgendwie involvieren,

Versucht nie, das alles gleichzeitig zu optimieren. Versucht immer, alle bis auf eine von diesen Metriken irgendwie rauszufinden, bis wohin ist es okay. Und dann, also meistens geht es dann um entweder Accuracy oder AOC oder sowas, ihr dann optimiert und alles andere versucht dazu, Randbedingungen zu machen, die ihr einfach erfüllt. Und alle Modelle, die es erfüllen, die bleiben im Rennen. Das ist hier alles nochmal erklärt.

Und hier ist diese Situation, wo der Chef kommt und mich nervös macht, weil irgendwie er noch eine andere Firma beauftragt hat. Und die Konkurrenz hat also höheren Recall geschafft als ich. Und der Chef findet das wichtig, weil er wahrscheinlich auch so eine interne Kostenmatrix hat. Habe ich hier gesagt, was das Modell macht?

Ich glaube, es war meine Annahme, dass es um das gleiche Szenario geht, also Network Intrusion. Hab ich es geschrieben? Ja, irgendwie ist klar, oder? Aus der Formulierung. Jedenfalls denkt der Chef sich natürlich, Recall ist wichtig, denn wenn wir einen Einbruch in unser Netzwerk übersehen, dann wird es richtig übel. Dann, was weiß ich, löschen die Hacker all unsere Daten oder

erpressen uns oder was auch immer. Davor hat er natürlich Angst. Das heißt, Recall findet er irgendwie wichtig. Kann man nachvollziehen. Jetzt die Frage, was antworten wir ihm? Oder was ist eigentlich unser Ziel? Was wollen wir erreichen? Und ich glaube, natürlich steckt hier wieder mein Träbel für Kostenmatrizen dahinter.

Man kann erstmal sagen, ja, ja, ich kann natürlich ein System mit perfektem Recall bauen. Ja, also wenn du mir jetzt sagst, ich soll auf Recall optimieren, dann baue ich einfach den umgekehrten Constant und sage immer Alarm. Das heißt, ständig sind wir am Schauen, ob es tatsächlich eine Network Intrusion gibt, weil das Modell ständig sagt, Achtung, Achtung, Achtung.

Einbruch ins Netzwerk. Dann haben wir super Recall, aber wir haben auch keine Zeit mehr, irgendwas anderes zu machen in der Firma, weil alle nur noch mit Network Intrusion beschäftigt sind. Dann wird der Chef vielleicht verstehen, okay, Recall ist nicht alles, wir müssen es irgendwie balancieren mit der Precision. Also, dass ständig alle nur noch in Alarm sind, resultiert ja aus einer sehr niedrigen Precision. Und

Wie finden wir die Balance? Und ich denke, es gibt vielleicht mehrere Wege, die Balance zu finden. Wir können auch meinetwegen uns auf das F-Measure einigen, als Metrik, die optimiert werden soll. Ich denke, in dem Fall kann man vielleicht auch eine Kostenmatrix aufstellen, wo man irgendwie erfasst, was kostet es, einen angeblichen Fall von Network Intrusion zu prüfen und zu gucken, ob es da wirklich ein Problem gibt.

Und was kostet es? Zugegeben, das wird eine sehr hohe Zahl sein, was kostet es, wenn wirklich Network Intrusion stattfindet? Da gibt es natürlich auch wahrscheinlich eine hohe Bandbreite, aber Hacker haben selten was Gutes im Sinn, da wird es schon teuer werden. Aber ich denke, na, wo ist es denn jetzt? Genau, man sollte einfach die Zeit nehmen, nochmal zu überlegen, was man eigentlich wirklich will und man muss irgendwie Precision und Recall balancieren. Vielleicht wird es das F-Measure sein am Ende oder eben

Die Kostenmatrix. Ja, Marc. Eben, also das war, was ich gesagt habe, über den Recall, also wenn ich wirklich nur versuche, alle Fälle von Network Intrusion zu finden oder dass mir keine entgeht, dann kann das dazu führen, dass ich ein Modell

präferiere, was einfach ständig Alarm schlägt. Also im schlimmsten Fall jede Minute. Hat einen super Recall, da wird nichts verpasst. Also vielleicht doch, weil die Leute irgendwann denken, lass mich in

Ruhe, ich gucke nicht mehr nach. Aber verstehst du, was ich meine? Also Recall allein ist nicht genug. Eigentlich nie. Wir müssen immer auf uns davor wappnen, dass es einfach viel zu viele Alarme oder viel zu viele

Ja-Vorhersagen gibt. Auch im Marketing. Wenn ich Targeted-Marketing machen will und ein Modell habe, was einfach immer Ja sagt, dann ist es kein Targeted-Marketing mehr. Dann ist es einfach Mass-Campaign. Irgendwie muss man immer die Balance finden. Und ja, die Kostenmatrix ist oft ein gutes Mittel dafür. Noch weitere Fragen? Zweifel? Okay. Gut. Dann machen wir jetzt typische Probleme.

ein paar folien mit nur und am ende wollen wir ein escape game spielen wird schwierig also passt gut auf das gute ist dass es vielleicht deswegen weniger schwierig ist weil es wieder auf diesem bank beispiel basiert was wir schon kennen so halt genau jetzt sind wir irgendwie immer noch da unten bei evaluation oder eigentlich vielleicht bei dem ganzen cycle

Und wir werden zwei Probleme uns anschauen. Wir werden einerseits Over- beziehungsweise Underfitting uns anschauen, also die Suche nach dem richtigen Level von Komplexität unseres Modells. Und andererseits das Problem der Nadel im Heuhaufen, also Class Imbalance. Entschuldigung, ich verwende immer die englischen Wörter, aber wir haben ja gesehen, dass es vielleicht manchmal besser ist. Wo ich eben bei Klassifizierung einen Wert des Klassenattributs habe.

der sehr, sehr, sehr viel häufiger ist als der andere. Und die anderen interessieren mich meistens. Also die Leute, die auf meine Kampagne reagieren, sind leider immer nur sehr wenige. Oder die Fälle von Network Intrusion, die eben sehr selten sind. Oder die Fälle von Fraud oder von Churn oder was. Okay, fangen wir an mit dem Over- oder Underfitting. Ich habe jetzt wieder hier so ein paar Begriffe, alle auf Englisch. Ähm,

Aber eben auch, wenn Leute auf Deutsch was schreiben, dann verwenden sie meistens diese Begriffe. Insofern würde ich sie eher nicht übersetzen wollen. Man spricht von, also wie gesagt, es geht darum, die Komplexität eines Modells an die Komplexität der Daten anzupassen. Wenn die Daten einfache Muster enthalten, dann sollten wir vermeiden, ein sehr, sehr komplexes Modell daraus zu lernen.

weil es meistens Dinge erfasst, die gar nicht da sind sozusagen, also die nur in den Trainingsdaten sind, aber nicht darüber hinaus. Das ist das, was ich Overfitting nenne. Und ein weiterer Begriff dafür ist Variance, also große Variance, warum auch immer. Nicht zu verwechseln mit der Varianz in der Statistik. Andererseits, wenn meine Daten sehr komplexe Muster enthalten, dann brauche ich auch ein Modell, was hinreichend komplex ist. Wenn ich es nicht habe, dann spreche ich von Underfitting, logisch.

Oder auch der Begriff Bias wird verwendet dafür. Also ein großer Bias. Ich sage das nur, damit ihr diese Begriffe kennt. Ihr werdet ihnen vielleicht noch begegnen irgendwo. Also fangen wir mal an mit dem Modell, was nicht komplex genug ist. Und das ist oft der Fall, weil ich gewisse Phänomene in meiner Trainingsmenge nicht gesehen habe, schlicht und einfach, weil die Trainingsmenge zu klein ist. Also hier zum Beispiel seht ihr ein Modell. Ihr könnt euch vielleicht wieder vorstellen, dass es um...

Kreditvergabe geht und das eine ist das Einkommen und das andere die Schulden oder so. Und wenn das hier zum Beispiel das Einkommen ist, dann sagt das Modell, das reicht mir. Ich schaue die Schulden gar nicht an. Ich sehe die armen Leute, die zahlen nie zurück und die mit dem hohen Einkommen, die zahlen zurück. Fertig, das ist mein Modell. Also dieses Modell benutzt jetzt nur eins von zwei möglichen Features und hat aber vielleicht einfach viele

Trainingsdaten hier nicht gesehen, also die nicht ausgefüllten roten Punkte. Und jetzt, wenn man sich das anschaut, dann würde man die Linie zur Trennung von den roten und den blauen vermutlich anders wählen. Also man würde vermutlich so ein Modell versuchen zu lernen oder man würde so eins lernen, wenn man die unausgefüllten roten Punkte als Trainingsdaten gehabt hätte. Und dieses Modell, ja, also ihr seht jetzt sozusagen, dass es schräg geht, also beide

beide Attribute benutzt, also auch die Schulden spielen jetzt eine Rolle. Also was könnte jetzt sozusagen der konkrete Fall sein, bei dem sowas auftritt? Erinnert ihr euch an die logistische Regression und an die Regularisierung? Ich komme da auch gleich nochmal drauf. Also wenn ihr jetzt hier aufmacht bei der logistischen Regression, dann seht ihr, hier könnt ihr zwei Arten von Regularisierung wählen, also

außer eben keine, gibt es Lasso und Ridge. Und ich glaube, ich hatte es mal erwähnt, es gibt es auch bei der linearen Regression, bei beiden Regressionsarten. Und Lasso ist was, wir hatten das gemacht, oder? Mit den Häusern. Erinnert ihr euch? Mit irgendeinem Datensatz, mit uns auch irgendwie im Häusern oder so. Jedenfalls, wenn ich auf Lasso setze dann vor allem, werden Gewichte auch gern mal null. Und wenn ich zum Beispiel, also...

Logistische Regression besteht ja genau aus so einer Linie. Und wenn ich Lasso einstelle, dann reduziert mir das die Komplexität meines Modells. Oft eben auch sogar dadurch, dass es Features ganz auf Null setzt oder die Koeffizienten der Features, also welche rausschmeißt. In dem Fall hier also die Schulden. Und das führt zu einem einfachen Modell. Aber in dem Fall ist es eben zu einfach gewesen, wenn ich...

auf Lasso verzichtet hätte, dann hätte ich vielleicht von Anfang an so eine Linie gelernt. Ja, man weiß es nicht. Also, wie gesagt, zwei Arten, wie man es teilen kann oder lösen kann. Mehr Trainingsdaten helfen oft oder eben ein komplexeres Modell. Hier steht es nochmal als Bullet Points. Und ich habe mal diese Tabelle gemacht und die Tabelle kommt dann beim Overfitting gleich wieder. Und

im Prinzip in gleicher Form, weil man einfach sieht, wie kann man die Komplexität steuern. Und vielleicht nochmal zur Erklärung, was bedeutet K und R. K steht für Klassifikation, R für Regression. Und die Algorithmen haben sozusagen per se erstmal eine Komplexität. Das heißt, ich kann, wenn hier steht komplexeres Modell wählen, entweder einfach einen anderen Algorithmus verwenden oder ich bleibe bei meinem Algorithmus und wähle eine dieser Algorithmen.

Optionen hier an oder ab. Also beim Underfitting würde ich das dann eher nicht machen. Also würde nicht regularisieren, sondern das mal größere und mehr Koeffizienten nicht bestrafen. Oder beim Baum habe ich auch den Parameter, der mir hilft, die Komplexität zu steuern, indem ich zum Beispiel die Tiefe begrenze. Gleich, Dennis, den Satz noch.

Genau, oder Gradient Boosting, habe ich auch die Tiefe der Bäume oder eben die Anzahl der Bäume, das sind meine Hebel. Bei Qualen hatten wir bei Rekursionen das Problem, dass wir dann eben in den Bäumen Bälle hatten, dass dann das noch ein bisschen umgehezt war, weil es dann viel zu tief geschätzt wurde. Ja, also...

Da hatten wir auch so ein Problem, dass wir zu wenige Trainingsdaten hatten aus einer bestimmten Region. Du meinst sozusagen die ganz großen Häuser, für die es keine ähnlichen in der Trainingsmenge gab. Und dann hatten wir den Preis von so einem Haus zu niedrig geschätzt. Genau, also sozusagen...

Das war ja kein komplexer Zusammenhang, der war eigentlich ziemlich linear. Aber es gab einfach zu wenig Trainingsdaten für den KNN. Es gibt auch die Kreuz- und Dörfer-Regressionen. Sind die zu interpretieren? Es eignet sich einfach für gleiche Teilen. Du kannst es für beide nehmen, genau. Das heißt nichts weiter, als dass du es benutzen kannst. Also ich wollte einfach sozusagen eine Folie machen, auf der ich sowohl Regression als auch Klassifikation abhandle.

Und da muss ich viele Algorithmen zweimal hinschreiben, weil sie einfach für beides verwendbar sind. Also ich kann mit keinem und beides machen. Okay, gut. Eben, wenn man Underfitting verstanden hat, dann ist Overfitting auch wahrscheinlich relativ schnell erklärt. Da geht es eben darum, das übertreibe mit der Komplexität. Also hier haben wir jetzt so quasi das Bild von vorher, wo sich das so schön separieren lässt, eigentlich durch eine

Linie, die immerhin beide Attribute benutzt, aber eine schöne Linie. Und wenn ich jetzt einen Punkt habe, den ich hier als Noise Point bezeichnet habe, also warum tue ich das? Warum nenne ich den Noise? Ich denke, wir würden wahrscheinlich übereinstimmen, dass wenn jetzt hier ein Punkt liegt, haben wir die Intuition, dass der auch rot ist. Der ist vermutlich einfach...

irgendwie eine Ausnahme hier in der Blau. In dieser Region, im Korridor hier, den das Modell jetzt bildet, wenn ich Oberflächen mache, würde es jetzt, also was dieses Modell macht, ist alles, was hier drin liegt, als Blau zu platzifizieren. Und vermutlich ist es aber nicht so. Vermutlich ist einfach dieser blaue Punkt so eine Art Ausnahme und die Punkte, die in dem Korridor liegen, sind ansonsten rot. Ich habe keinen Beweis dafür jetzt, aber in vielen Fällen ist das so.

Es gibt einfach nicht die perfekte Linie. Aber in dem Fall ist es besser, die Linie zu nehmen, die nicht ganz perfekt ist, als diesen Korridor. Weil wenn wir jetzt mehr Trainingstagen aus dem Bereich sehen würden, würden wir dort vermutlich viele Fehler mit diesem Modell machen. Wenn das stimmt, was ich sage. Zweites Bild.

zeigt, was man dann manchmal beobachtet. Also, was habe ich hier abgetragen? Ich habe auf der x-Achse die Komplexität eines Entscheidungsbaumsmodells einfach gemessen in der Anzahl Knoten, die der Baum hat. Also zum Beispiel, je tiefer ich den Baum mache, desto mehr Knoten bekommt er. Wächst relativ schnell die Anzahl der Knoten mit der Tiefe. Und die rote Linie, also was wir auf der y-Achse haben, ist der Fehler, der Klassifikationsfehler. Also sozusagen $1 - \text{Accuracy}$.

Und wenn wir die rote Linie anschauen, dann ist das, was wir kriegen, wenn wir auf dem Trainings-Set evaluieren. Ich habe euch letztes Mal gezeigt in Orange, man kann das einstellen. Also muss ich aufpassen. Doch, ihr seht, sehr gut. Ich kann hier sagen, Test on Train Data. Er rechnet noch. Und ich hatte gesagt, es ist eigentlich nicht erlaubt. Also es ist keine richtige Art, sich

einen Eindruck zu verschaffen, wie gut das Modell auf neuen Daten sein wird. Weil wir ja nur auf den Daten evaluieren, die das Modell schon bestens kennt. Aber ich kann es eben nutzen. Also ich sehe jetzt hier zum Beispiel die logistische Regression hat 90% Accuracy auf den Trainingsdaten. Und wenn ich jetzt umstelle auf Random Sampling, dann ist es fast genauso gut. Also habe ich da kein Overfitting-Problem.

Beim Tree, lass uns mal schauen, wie war es da noch? Wenn es nicht so lange dauern würde, könnte man das da mal hin und her switchen. Hab ich schon wieder vergessen. Ich glaube, ich mach's jetzt nicht nochmal. Jedenfalls, wenn ich sozusagen switche zwischen der Evaluation auf echten Testdaten und der Evaluation auf Trainingsdaten und ich sehe sehr große Unterschiede, dann weiß ich, ich hab ein Overfitting-Problem und

Anhand dieser Folie sieht man auch, dass es quasi so den richtigen Punkt gibt, was die Komplexität des Modells anbelangt. Also natürlich, ich kann es immer komplexer machen und werde dadurch immer besser mich diesen Trainingsdaten anpassen und der Fehler wird immer kleiner. Aber auf neuen Daten merke ich dann, dass ich Sachen gelernt habe, die nicht generalisieren. Also die auf den Trainingsdaten funktionieren, aber nicht auf den ungewohnten Daten.

Und das ist so ungefähr der Punkt hier bei 125, 130 Knoten und dann die steigende Kompetenz zu eigentlich schlechteren Ergebnissen, also höheren Fehler auf der Testmenge führt. In Orange meinst du? Ja.

Ich glaube nicht, nein. Du müsstest es von Hand. Nein, kannst du nicht. Ich weiß. Also du kannst es natürlich von Hand machen. Also du kannst mit verschiedenen Tiefen und dann jeweils messen und es selber abtragen. Oder du nimmst Python. Dann kannst du natürlich eine Schleife machen. Schleifen kennt ihr schon, oder? Ja.

und die Tiefe variieren und jeweils messen und dir das ausgeben lassen. Also es muss ja auch Gründe geben, warum man noch Python macht. Ja, also du hast recht, es wäre ultra hilfreich, aber es gibt wahrscheinlich viele Sachen, die ultra hilfreich wären und irgendwo waren die Entwickler von Orange, dann haben sie gefunden, jetzt ist gut. Funktioniert Test und Testing? Machen wir heute. Du wirst es dann sehen.

Ich habe es schon vorbereitet, ihr müsst es nicht selber machen. Ich hatte gesagt, Test-on-Test-Data bedeutet, dass ihr, ihr könnt zum Beispiel das verwenden, wenn ihr selbst euren Split machen wollt. Und ihr werdet nachher verstehen, warum es manchmal wichtig ist, selber zu kontrollieren, wie die Daten in Trainings- und Testmenge aufgesplittet werden. Zum Beispiel, weil man kontrollieren will, dass irgendwas immer auf der gleichen Trainingsmenge passiert. Das ist das, was wir nachher machen wollen. Manchmal hat man aber auch schon von vornherein irgendwie

eine Testmenge zur Seite gelegt in einem extra File, irgendwie außerhalb von Orange und dann kann man die reinladen und als weiteren Input vom Test-and-Score-Widget verbinden und dann Test-on-Test-Data und man muss dann sozusagen, also wenn ich jetzt, ich kann es mal zeigen, jetzt mangelt es besserer Ideen, würde ich jetzt einfach nochmal die gleichen Daten laden, das ist dann wie, wenn ich auf den Trainingsdaten evaluiere, aber

Du musst dir jetzt vorstellen, dass ich noch weitere Bankkunden ausprobiert habe, angerufen habe und geguckt habe, ob sie reagieren. Ich nehme jetzt einfach noch die gleichen Daten. Und wenn ich das jetzt hier als weiteren Input verbinde, er macht es jetzt automatisch, dass er hier die Rolle festlegt. Also wenn Widwit viele Inputs hat,

viele mögliche, dann ist es manchmal so, dass man die Rolle festlegen muss und es steht dann hier am Pfeil dran. Also diese Daten spielen jetzt die Rolle von Testdaten und die, die hier reinfließen, einfach von Daten, also in Klammern Trendingsdaten. Und wenn ich jetzt hier auf Testdata umstelle, dann funktioniert es nicht, warum auch immer. Ja, tatsächlich habe ich damit schon öfter Probleme gehabt, gerade bei Test und Score. Wir werden es nachher mit dem Predictions Widget machen. Eigentlich habe ich es aber auch schon mal geschafft.

Ja, ich glaube, ich finde das jetzt gerade nicht raus, was das Problem ist. Ach so, ja, ja, nee, das kriegen wir hin, warte mal. Wir müssen einfach auch hier, ja, so, dann klappt es vielleicht, mal schauen. Ja, jetzt klappt es. Okay. Manchmal ist es ja so, kaum macht man es richtig, schon geht es. Müssen wir nicht abwarten, oder? Ah, da. Also das ist jetzt in dem Fall, die Zahlen werden die gleichen

sein, wenn du Test- und Train-Data auswählst, weil ich jetzt einfach nochmal die Trainingsdaten geladen habe, aber eben, stell dir vor, ich habe andere Daten.

Weiter? Pause? Wartet mal, lass mal gucken, was haben wir noch? Ich meine auch bis da, okay? Dann hat es mehr Sinn. Geht schnell. Genau, die Diagnose habe ich euch schon gezeigt. Also Overfitting kann ich im Gegensatz zum Underfitting besser feststellen. Also Underfitting, wie stelle ich das fest? Oft indem ich einfach Komplexität erhöhe und gucke, ob es besser wird. Das kann ich natürlich beim Overfitting auch machen. Aber da ist es noch einfacher, da kann ich eben einfach

im Test- und Score-Widget in Orange zum Beispiel umstellen, Test- und Train-Data und dann, wenn es sehr viel besser wird, also es wird immer besser, aber wenn der Unterschied gewaltig ist, also ich habe, wenn ich jetzt mal Accuracy nehme, vielleicht 70% Accuracy auf den Testdaten und 95% Accuracy auf meinen Trainingsdaten, dann weiß ich, aha, das riecht nach Overfitting. Und was kann ich tun? Also das habe ich, glaube ich, auch schon mal gezeigt. Underfitting links und Overfitting rechts.

Und ja, meistens, wenn ich es komplexer mache, das Modell, dann laufe ich mehr in die Overfitting-Gefahr rein. Und wenn ich es einfacher mache, dann... Also das ist immer der Trade-off, mit dem man kämpft. Und logischerweise ist das, was man tun kann, im Prinzip das Gleiche wie beim Overfitting, eben einfach umgekehrt. Also entweder eine...

Arbs von Modellwellen, die hier eine geringere Komplexität hat oder beim gleichen Algorithmus bleiben und wieder diese Stellschrauben nutzen, also zum Beispiel beim Baum eine kleinere Tiefe vorgeben oder eben regularisieren, weniger Bäume für Spray-Dent-Boosting und so weiter. Was man da oft macht auch, dass man sagt, man hat

wenn man jetzt zum Beispiel einen Baum lernt und das machen will, was du gerade gesagt hast, Dennis, was ja doch schön wäre, wenn es eingebaut wäre sozusagen, dass man sagt, okay, ich nehme mir einen Teil meiner Daten als Trainingsmenge und dann nehme ich eine sogenannte Validierungsmenge und die Testmenge. Und die Validierungsmenge, die nutze ich sozusagen, um hier die Tiefe des Baumes zu bestimmen, die am besten ist. Also da kann ich auch dieser Validierungsmenge

die blaue Kurve zeichnen, die wir hier hatten. Also da spielt sie dann die Rolle des Test-Tests und ich sehe sozusagen, ab welcher Knotenanzahl es wieder schlechter wird, also ab welcher Komplexität und dann einige ich mich auf diese Tiefe sozusagen und das Test-Test ist sozusagen die reine Lehre, sagt, dass man immer noch Daten übrig behalten soll, die man noch für gar nichts verwendet hat, für keine Art von Tuning, um zu sehen, wie gut das Modell wirklich ist.

Genau, und das Validierungs-Set nennt man auch manchmal Dev-Set in der Englischen. Also wenn ihr so im Internet stöbert, dann sprechen viele Leute von Dev-Set. Genau, jetzt wäre der Moment, falls es nicht noch eine Frage gibt. Gut, dann zählt es sagen bis um 25. Machen wir Pause.

... ..

Vielen Dank.

So, jetzt kommt... Also...

Das zweite typische Problem, übrigens, jetzt wo ich gerade dran denke, gehe ich hier nochmal raus, wo ich bei typischen Problemen bin, Teams nervt heute aber auch, ich habe euch noch ein Buch hier hochgeladen, oben, bei Allgemeines, Machine Learning Design Params,

Da geht es nicht um Probleme vordergründig, aber die, also es geht um Lösungen, aber zu jeder Lösung gehört natürlich ein Problem. Und da sind echt gute Tipps dabei. Und die Sachen, die wir hier besprechen, zum Beispiel Rebalancing, ich nenne es dann anders, aber es geht im Grunde um das Gleiche, steht hier auch drin. Und das mit dem Overfitting, vielleicht gibt es noch Useful Overfitting,

Ja, weiß gerade nicht, wo es ist. Es sind sozusagen typische Probleme und ihre Lösung noch sehr viel mehr, als wir hier besprechen. Aber es kann ein Begleiter werden, wenn ihr tiefer einsteigt in manche Sachen später. Jetzt erstmal eins von den wirklich häufigen Problemen, die Class Imbalance. Das kennen wir ja schon, also dieses Nadel im Heuhaufen Problem, wie ich es genannt habe. Und die vielen Beispiele habe ich auch schon erwähnt. Also bei Marketing habe ich

Typischerweise eine sehr geringe Antwortrate. Hoffentlich sind die meisten meiner Schadensfälle in der Versicherung kein Fraud oder die meisten meiner Kreditkartentransaktionen kein Fraud oder was auch immer. Also Fraud sollte hoffentlich auch selten sein. Auch wenn ich zum Beispiel, das wäre jetzt hier ein Beispiel aus der Medizin, Patienten beobachte und jeden Tag soll mein Modell mir vorhersagen,

ob es ein Risiko gibt für was auch immer, zum Beispiel Spitaleinweisung für den nächsten Monat oder so, dann ist auch da hoffentlich selten der Fall, dass es mich davor warnt oder dass es mich davor warnt, dass Kunden mich verlassen. Also Churn sollte hoffentlich auch selten sein. Also wir haben ganz viele solche Beispiele und wir haben schon gesehen, Accuracy ist dabei problematisch. Also vielleicht erinnert ihr euch an die Mail-Magazines. Und wenn ich Accuracy nehme,

und ein Modell betrachte, was immer No sagt, dann hat das eben ein paar 90% Accuracy und scheint sehr gut, dabei ist es völlig sinnlos. Okay, jetzt habe ich das hier schon verraten, warum nochmal. Alternativen kennen wir, insbesondere bei den Kosten, ist der, bin ich Fan,

Und wenn wir Precision Recall und F-Measure verwenden, dann in Orange oben bei Evaluation Results for Target immer Yes einstellen. Also angenommen, dass die Yes-Fälle eure Nadeln sind und die No-Fälle der Heuhaufen. Genau, jetzt gibt es zwei Strategien, die man typischerweise verwendet, um Modelle, ja, ich sag mal, sensibler zu machen für die Nadeln. Also wenn ich von Marketing spreche, dann

wäre das Modell optimistischer. Wenn ich von Fraud spreche, dann möchte ich eher eins, was ein bisschen pessimistischer ist, also häufiger Fraud vorhersagt. Also in allen Fällen möchte ich ein Modell, was häufiger die Minderheitsklasse vorhersagt, also die Nadel. Einerseits manchmal, um die Chancen zu ergreifen, die zum Beispiel beim Marketing darin

in den Nadeln liegen oder eben die großen Risiken zu vermeiden, die von den Nadeln aufgehen, zum Beispiel bei Fraud. Und Undersampling, und man muss immer dazu sagen, Undersampling, die Majority Class, also die Mehrheitsklasse, verkleinern. Also ich habe euch ein Buch gerade gezeigt, Rebalancing ist ein Wort für beide Verfahren hier. Ich Sorge irgendwie dafür, dass dieses Ungleichgewicht

kleiner wird oder ganz verschwindet. Hier habe ich das so gemacht, dass es ganz verschwindet. Also dass ich am Ende so viele Nadeln wie Heuhaufen habe. Und hier mache ich das beim Undersampling, indem ich einfach Heu weglass. Also in dem Fall vermutlich zufällig einfach nur ein Sample von der Mehrheitsklasse nehme, solange in diesem Fall bis es gleich groß ist wie meine Nadeln. Kann natürlich auch hier mit dem Verhältnis spielen. Also ich muss es nicht

Ich kann es sogar noch redbar machen oder eben jede mögliche Mischung ist erlaubt. Also es gibt da kein richtig oder falsch. Am Ende, wenn ich zum Beispiel kostenbasiert evaluiere, dann finde ich den Grad an der Sampling, der zu den besten Kosten führt. Und umgekehrt kann ich Oversampling betreiben, das im einfachsten Fall so geht, dass ich einfach die Nadeln

Vervielfache. Also hier habe ich es jetzt nicht ganz gleich groß gemacht am Ende. Ich habe hier einfach die Nadel portiert und somit ist es sehr viel besser balanciert als vorher, aber nicht perfekt. Also da hätte ich jetzt vielleicht es versechsfachen müssen, dann wäre es wirklich gleich groß geworden. Manchmal macht man da auch die

Papier-Training nicht einfach durch Kopieren, sondern indem man auch synthetische Beispiele erzeugt. Also man nimmt die originalen Nadeln und dann nimmt man da so kleine zufälligen Veränderungen dran vor, sodass es dem Algorithmus erscheint, als wären es wirklich neue Daten. Und ja, im Wesentlichen aber stützt man sich in den ganzen Verteilungen und Eigenschaften der Nadeln sozusagen auf die Original-Training-Star und sorgt dafür, dass es mehr von der gleichen Sorte werden kann.

Und was wollte ich noch sagen dazu? Es gibt auch die Möglichkeit, beides zu kombinieren. Also es gibt zum Beispiel, das müsste eigentlich in dem Buch drinstehen, mal kurz nach SMODE suchen. Synthetic Minority Oversampling Technique. Da werden Beispiele synthetisiert. Also SMODE, ich habe mal angefangen, das zu recherchieren für Orange.

wird nicht angeboten, weil es Entwickler gibt in der Orange Community, die das für großen Schwächen halten. Das ist nicht meine Meinung, aber das ist der Grund, warum es das nicht gibt. Man kann aber sowohl Undersampling als auch Oversampling relativ leicht selber basteln in Orange mit ein, zwei Widgets und das machen wir nachher im Escape Game. Deswegen prägt euch ein, wie das hier aussieht.

Ihr müsst einfach, wenn ihr zum Beispiel Oversampling machen wollt, irgendwie dafür sorgen, dass die Nadeln vervielfacht werden und dann wieder gemischt werden mit dem Heu. Oder mit den ursprünglichen Daten. Sagen wir es mal so. Okay, ist das soweit klar, wie es funktioniert? Also wenn ihr ein Modell auf diesen modifizierten Datensätzen trainiert, dann wird es automatisch sensibler für die Nadeln. Und wenn ihr zum Beispiel...

sagt, okay, wenn man Nadeln findet, dann führt es zu einem besonders hohen Gewinn und der ist viel höher als der Verlust bei einem Falls Positive und habt das in der Kostenmatrix festgehalten, dann wird das auch ziemlich sicher dazu führen, dass die Kosten sinken. Eigentlich müsste eine Frage jetzt sich auch dringen. Wenn ihr jetzt diese zwei Varianten seht, fragt ihr euch, was ist besser und warum? Also natürlich kann man es nicht so einfach sagen, sonst hätte ich euch ja nur das eine gezeigt.

Also es gibt nicht die Antwort, muss man ausprobieren. Es gibt eine Tendenz. Die Tendenz ist, dass wenn man von Anfang an sehr wenige Daten hat, man eher nicht anders Sampling macht, weil man dadurch ja auch noch mehr Daten verliert. Da funktioniert Oversampling meistens besser. Wenn man eine größere Menge an Daten hat, dann haben die Leute mal diese, mal jene Erfahrung gemacht. Also ausprobieren, so wie so oft im Maschinenlanding.

Genau. Eine Sache ist wahnsinnig wichtig bei der Sache. Man muss innerlich sehr aufpassen, dass man dieses Rebalancing, also entweder weniger Heu oder mehr Nadeln produzieren, nur beim Trainieren macht. Das ist eigentlich ein generelles Prinzip. Ihr könnt euch quasi erlauben, was ihr wollt, während ihr ein Modell trainiert. Also ihr könnt

Daten dazuholen von irgendwo anders, die eigentlich ganz anders sind als die Daten, mit denen ihr eigentlich arbeiten wollt. Solange es hilft, ein besseres Modell zu lernen, tut es. Aber niemals verwendet irgendwas, was nicht eurem eigentlichen Szenario entspricht, als Testdaten. Das heißt, das hier bitte nur auf den Trainingsdaten machen. Und deswegen werden wir auch

nachher wirklich erst, bevor ihr irgendwelches Oversampling macht, ein Sample von Daten zurücklegen, welches diese ursprüngliche Verteilung auch weiß, auf dem dann getestet wird. Also Testdaten werden wir uns beiseite legen, die natürlich dann diese Verteilung auch weiß. Und dann erst für die Trainingsdaten, die übrig bleiben, fangen wir dann vielleicht an, Heu wegzuschmeißen oder Nadeln zu vervielfachen. Nicht auf den Testdaten. Und wir testen dann wieder

auf Daten, die so aussehen wie da rechts oben. Und das bedeutet natürlich, wir haben jetzt dem Modell sozusagen mehr Mut eingeflößt, Ja zu sagen. Und das bedeutet zum Beispiel, dass die Accuracy sinken wird. Also es wird vor allem viele false positives produzieren. Aber wir wissen ja innerlich, in vielen Fällen ist das nicht so schlimm. Also die Kosten für die false positives sind hoch.

im Vergleich zu dem Gewinn durch True Positives oder Kosten für False Negatives meistens eher gering und deswegen lohnt sich der Mut. AOC sollte eigentlich in den meisten Fällen steigen, vermutlich manchmal nicht sehr stark. F-Measure sollte auch steigen, auch das vielleicht manchmal nicht so stark, weil es ja auch die Precision stark beinhaltet und

Wenn zwar der Recall steigt, aber die Precision im gleichen Maße sinkt, dann ist das vielleicht gar nicht so gut. Aber tendenziell ist das der Effekt, den wir uns erhoffen. Insbesondere, und da bin ich mir relativ sicher, wenn es einigermaßen funktioniert mit dem Over- oder Undersampling, dann sollten die Kosten sinken. Also entweder im positiven Bereich kleiner werden oder noch negativer, also mehr Gewinn. Genau, also wir erwarten mehr False Positives in der Confusion Matrix.

Aber eben auch mehr True Positives. Das ist das, warum wir es ja überhaupt erst machen. Soweit? So unklar? Okay. Jetzt sind wir schon nah am Escape Game, aber noch ein paar Tipps. Also, was kann man tun? Wie gesagt, Kosten, da muss man auch reden. Also, wenn man verstehen will, welche Auswirkungen es hat. Also, es fängt damit an, wenn ihr euch erinnert beim letzten Mal.

Ich hatte gesagt, wenn man eine Kostenmatrix aufstellt, muss man erstmal wissen, was wird eigentlich, was wird man tun, wenn das Modell Ja sagt, was wird man tun, wenn das Modell Nein sagt. Das muss man auch genau verstehen. Also wir hatten das bei der Versicherung, da haben wir gesagt, eigentlich gibt es zwei Möglichkeiten. Entweder wir schreiben direkt einen Brief an den Kunden und sagen, sorry, wir bezahlen das nicht und warten dann und klären es vor Gericht oder wir lassen es noch vorher jemandem anschauen.

Im Prinzip ist beides möglich. Das heißt, man muss erstmal rausfinden, was wird überhaupt passieren und dann, was wird es kosten, wenn man dies oder jenes tut. Dann hatten wir schon gesprochen, dass man immer versucht, nur eine Metrik zu finden, die man optimieren will und dann andere als Randbedingungen zu verwenden. Das kennen wir ja schon. Das ist eigentlich das Gleiche wie die andere Folie. Insofern habt ihr die Folie jetzt doch. Ich habe auch vergessen, es zu machen. Ich mache es nachher. Die letzte Folie vom anderen Sliding.

Was mir noch wichtig ist, ich glaube, es ist die letzte Folie. Kannst du sehen? Warte mal. Nee, doch nicht. Passt. Vielleicht mache ich erst die letzte Folie, weil das habe ich auch gerade gesagt. Insofern passt es noch ganz gut rein. Ich habe gesagt, auf der Trainingsmenge könnt ihr machen, was ihr wollt. Manches hilft nicht, dann lasst es lieber bleiben. Aber letztlich ist alles erlaubt, was hilft. Also zum

Beispiel sich noch Daten besorgen, von denen man glaubt, dass sie irgendwie ähnlich sind zu dem, was man eigentlich braucht.

im Anwendungsszenario antreffen wird. Oder eben rebalancing oder hier habe ich gesagt resampling. Also macht, was ihr wollt. Wichtig ist eben, dass die Testmenge und aber auch, falls ihr so eine Validierungsmenge verwendet, dann sollte auch die so aussehen wie das, was ihr dann in der Realität antreibt. Denn auch hier, wenn ihr irgendwas tunet oder so, dann sollte sich das an dem orientieren, was am Ende wirklich gefragt wird.

Insbesondere eben, wie dieses Klassenattribut verteilt ist. Also wenn es nur sehr wenige Nadeln gibt, dann sollte sowohl in Validierung, lässt sich das so sein, dass es nur sehr wenige Nadeln gibt. In der Trainingsmenge kann es ganz anders sein. Aber auch bei anderen Attributen. Also wenn ihr jetzt Daten aus anderen Quellen dazu nehmt, dann werden sich vielleicht bei gewissen Attributen ganz andere Verteilungen aufweisen. Vielleicht ist das egal. Steigert trotzdem die Performance, aber immer validieren und testen auch.

der ursprünglichen Verteilung. Genau, und jetzt noch dieses, mein letzter Hinweis, es wird oft unterschätzt, meiner Erfahrung nach, die sozusagen die Mächtigkeit von guten Fehleranalysen. Also ich habe da ja auch

eine Demo, die ich euch zeigen kann. Ich hoffe, die verwirrt euch nicht. Also fangen wir mal einfach an. Ihr könnt Bias und Variants, also oder Over- und Underfitting versuchen zu analysieren. Also erinnert ihr euch noch? Wie finde ich raus, ob mein Modell an Overfitting leidet? Wie war nochmal dein Name? Nikola. Wenn die Tester... Wenn die Tester... Overfitting. Sag doch mal, was du gesagt hast.

Also grundsätzlich, wenn die PS-Daten versagen, dann geht das eigentlich bei der Anfälle zu. Und man erkennt dann, dass das Ding besser Ergebnisse mit den Trainingsdaten erzielt hat. Genau. Also insbesondere, wenn der Unterschied sehr groß ist. Also wenn es sehr viel besser auf den Trainingsdaten performt, als auf den PS-Daten.

Wie finde ich einen hohen Bias, also hohes Underfitting? Ja, also das geht eigentlich nicht, außer ich vergleiche es halt mit komplexeren Modellen. Also im Prinzip hatten wir das bei unserem Skigebiet. Da hatten wir mit einer linearen Regression angefangen. Und da habe ich euch schon gesagt, dass es möglich ist, einen Fehler auf ungefähr 1000 runterzudrücken. Und dann habt ihr gesehen...

nachdem wir das mit dem Monat noch geklärt hatten, dass man mit der linearen Regression nur auf 1500 kommt. Hättet ihr auch rausgefunden, wenn ihr noch mit anderen Modellen experimentiert hättet, dass das eben nicht optimal ist. Und das hat man dann gut verstanden, warum die lineare Regression nicht komplex genug war. Weil sie insbesondere diese Abhängigkeiten zwischen den Variablen nicht einbeziehen. Also einfach ausprobieren, so wie du auch. Also Fehleranalyse,

Das kann man erstmal anschauen. Und dann kann man vor allem auch, würde ich mir immer die Confusion Matrix anschauen, also nicht nur Precision Recall, das gibt meinen Eindruck, aber es ist noch plastischer, wenn man wirklich sieht, wie viele False Positives, wie viele False Negatives. Und manchmal gibt es da einen klaren Trend, also dass ein Modell eben sehr viel mutiger ist als ein anderes, also insofern sehr viel mehr False Positives produziert als das andere. Und entsprechend weniger False Negatives in den meisten Fällen.

Und dann hilft es auch, wenn man sich einfach mal von den Fehlern, die ein Modell macht, ein paar anschaut. Also manchmal kann man dann erkennen, zum Beispiel, was für eine Art von

Trainingsdaten dem Modell fehlen. Also dass vielleicht gewisse Fälle ein bisschen unterrepräsentiert sind und dass es mehr Beispiele dafür braucht. Und dann kann ich versuchen, mir diese Beispiele irgendwie zu besorgen. Und das, was ich hier als Demo habe, ist dieser Punkt hier. Also man kann...

Eben, da habe ich jetzt ein bisschen Angst, dass ich euch verwirre. Schaut einfach mal, was ich mache und wenn ihr es nicht versteht, dann ist es auch egal. Aber ich will euch diese Idee einmal gezeigt haben. Ich glaube, oft hilft es auch oder reicht es, wenn man sich einfach Beispiele von Fehlern anschaut und dann mal durchgeht und versucht zu verstehen, was ist da schiefgegangen. Und dann kann man schon auch ein bisschen zählen. Also ich würde mir vielleicht so 50 Fehler anschauen und dann, wenn ich sehe, es gibt zum Beispiel 25 von diesen Fehlern, haben den gleichen Grund.

und der Rest verteilt sich über andere Gründe, dann würde ich natürlich erst versuchen, diesen Grund irgendwie zu eliminieren. Also so ein bisschen priorisieren, was ich als erstes an meinem Modell tune. Ups, sorry. Jetzt machen wir das. Ich probiere es mal. Also die Idee ist, dass ich auf meiner Testmenge ja weiß, welche Dinge falsch vorhergesagt wurden von meinem Modell. Und jetzt könnte ich wieder hingehen und ein Modell lernen, was vorhersagt,

ob mein erstes Modell falsch vorher sagt oder nicht. Das heißt, ich versuche zu verstehen, was sind die Muster, die typischen Situationen, in denen mein erstes Modell Fehler macht. Versteht ihr die Idee? Und wenn ich diese Muster verstanden habe, dann ist das so gut, wie wenn ich das hier so manuell mir anschau. Also hier bin ich davon ausgegangen, eben kleine zufällige Stichprobe, vielleicht so 50 Stück oder so,

Meistens gibt es schon einen guten Einblick. Aber hier kann ich natürlich auch mit sehr großen Daten. Also je mehr, desto besser eigentlich. Ich muss halt, wenn mein Modell wenige Fehler macht, dann ist die Stichprobe eher klein, aber dann bin ich sowieso ja schon happy. Also ich zeige euch das mal. Ich habe den Workflow schon gebaut. Wir müssen ihn nur verstehen. Was ist denn das jetzt hier? Ich glaube, es sind wieder unsere Bankdaten. Ich versuche mal hier ein bisschen aufzuräumen. Genau, also kennt ihr, ja?

wir wollen Leuten, Leute überreden, in langfristige Einlagen zu investieren. Jetzt hier habe ich einen Data Sampler. Der wird auch gleich im Escape Game vorkommen. Und zwar lasse ich den 20% der Daten zufällig wählen und schicke diese Daten hier hin. Also für diese Daten, also ich glaube, es sind insgesamt ein bisschen mehr als 4000 Kunden in dem Datensatz. Es wären dann also ungefähr 800, die ich

bei Seite lege sozusagen und auf denen ich dann Predictions mache. Und für die analysiere ich dann nachher was für Fehler. Also auf diesen 800 werde ich dann verstehen, mal gucken, ob das stimmt mit den 800, ja, 824. Das ist sozusagen meine Testmenge. Okay, und hier habe ich dann 80%, das sind die Remaining Data, also logischerweise 80% meiner Daten, die ich nutze, um einen Tree zu bauen. Ich kann mir auch anschauen, wie der aussieht. Okay, keine Ahnung.

Man sieht, okay, unter bestimmten Umständen sagt er Interesse voraus. Okay. Hier kann ich auch sehen, die Predictions, das sind 824 Stück, wo ich jeweils sehe, was der Tree sagt und was richtig gewesen wäre. Man sieht schon da, dass der Tree eher pessimistisch unterwegs ist. Vielleicht können wir ihm mehr Optimismus einflößen später.

Aber jetzt erstmal erwarten wir also relativ viele Faults, Negatives, richtig? Ja. Okay, jetzt kommt hier ein Formel-Widget. Also, soweit ist noch klar, oder? Okay. So, was passiert denn jetzt hier? Also,

eigentlich, ihr seht ja, die neue Variable, die ich mit dieser Formel kreierte, heißt Fehler. Also, das ist hier ein Ungleichzeichen, ja? Also, Ausrufzeichen ist gleich, steht für Ungleich.

Und Tree 3, das hatten wir gerade in diesem Data Table hier gesehen, das ist das hier, diese Spalte. Das heißt, das ist die Vorhersage des Baums. Und wenn die Vorhersage des Baums nicht dem Y entspricht, und das habe ich hier noch als String verwandelt, dann sonst schmeißt es mir irgendeinen Fehler. Das heißt, wenn die, falsch ist auf gut Deutsch, die Vorhersage von dem Tree,

dann jetzt sonst noch okay ja und was ich jetzt mache ist also könnt ihr hier sehen sozusagen der tree sagt eigentlich wäre es jetzt gewesen dann muss hier hinten bei fehler jetzt stehen und da wo es übereinstimmt ok gut jetzt haben wir also diese neue spalte fehler und die schmeißen wir jetzt hier rein um sie vorherzusagen aus den anderen jetzt muss das mal ausprobieren

Okay, ich lerne also einen Tree und der sagt mir vorher. Und jetzt muss ich verstehen, dass die Blätter haben wieder Yes und No. Aber das bedeutet jetzt Fehler, Ja oder Nein. Und jetzt wird es ein bisschen komplex. Also hier zum Beispiel sagt das Modell, also dieses Modell, dass mein erster Tree einen Fehler macht. Und hier oben steht, dass du es dann gleich jetzt willst. Das heißt, ich kann auch sagen, was das für ein Fehler ist.

Es ist ein False Negative. Weil Y ist Yes. Und da es ein Fehler ist, muss mein Modell wohl No gesagt haben. Sonst wäre es ja kein Fehler. Also mein Modell hat No gesagt. Und in Wirklichkeit ist es Yes. Und ich sehe hier, auf dieser Seite des Baumes Fehler 78, die vorhergesagt werden. In Wirklichkeit ist es natürlich anders, die wirklichen Fehler.

Und da gibt es sozusagen ein relativ einfaches Muster. Also wenn ich mich wieder an der Größe der Zahlen orientiere, dann sagt es, okay, false negatives, bei denen num successes, also die Anzahl von vorher erfolgreichen Kontakten mit diesem Kunden, gleich null ist. Also alle Kunden, die ich noch nie erfolgreich überzeugt habe für irgendwas,

Die werden auch diesmal nicht überzeugt, sagt mein Modell. Also ich habe ja gesagt, false negatives, mein Modell sagt no, aber da gibt es 73, wo es eigentlich doch eine Reaktion dann gab. Also wir können auch hier sozusagen sehen, ursprünglich gibt es insgesamt 78 false negatives. Also hier das Modell sagt no, aber eigentlich wäre es jetzt. Das sind die, die man in dem Tree auch gut sieht, 73 plus 5.

Und jetzt kann man dann, ja, was heißt das jetzt? Jetzt kann man dann anfangen zu überlegen, ob man daraus irgendeine Strategie ableiten kann. Also für den ursprünglichen Baum, um die Anzahl der Fehler zu reduzieren. Also ich hätte jetzt zum Beispiel die Idee gehabt, dass man sagt, ah, das scheint mir ein bisschen zu einfach zu sein. Also einfach immer zu sagen, wenn es noch nie was gebracht hat, die Person zu kontaktieren, dann wird es auch diesmal nichts bringen. Vielleicht muss ich

die Komplexität erhöhen an der Stelle. Also dem Baum erlauben, noch etwas tiefer zu werden. Der ist ja hier begrenzt gewesen. Auf die Tiefe 4. Wenn ich jetzt mal auf 5 gehe, dann vermeide ich vielleicht ein paar Falls negatives. Also statt 78 habe ich jetzt nur noch 58. Kann auch Zufall sein. Also was das dann bedeutet und was man daraus schließt, ist jeweils natürlich wichtig.

individuell und nicht vorherzusehen und manchmal vielleicht auch falsch, aber es gibt mal, es kann mal einen Hinweis geben. Man kann es so machen. Wie gesagt, ich finde es auch genauso wertvoll, oft noch wertvoller, wenn man einfach sich mal ein Sample von falsch klassifizierten Instanzen nimmt und die sich einfach im Detail anschaut. Okay. Jetzt wären wir bereit für das Escape Game.

wo sich so ein Oversampling drehen soll. Also wir nehmen eigentlich wieder die gleichen Daten. Wir haben ja beim letzten Mal oder auch heute wieder festgestellt, eigentlich hätten wir vielleicht eine Kostenmatrix aufstellen sollen. Das wird in dem Game auch noch passieren, dass ihr das machen müsst. Und dann werdet ihr versuchen, in Orange eine Lösung zu bauen, die das Modell noch ein bisschen optimistischer macht. Fühlt ihr euch bereit?

Die Pause ist noch nicht so lange her, dass wir schon wieder eine bräuchten, oder? Ich weiß nicht, wie wollen wir es machen? Das wird eine Weile dauern, das Game. Also, wir können auch einfach Floating Break, nenne ich das manchmal. Also wir sagen einfach, ich muss sowieso schauen, wie lange ihr braucht. Aber wenn ihr jetzt zwischendurch mal 10 Minuten Pause braucht, dann nehmt sie euch an. Nutzt den Platz, arbeitet im Gruppenbereich,

Wenn ihr zu zweit seid oder zu dritt, meinetwegen auch, könnt ihr nebeneinander sitzen. Sonst würde ich empfehlen, um einen Tisch rum sich zu scharen und dann sich gegenseitig zu inspirieren. Jetzt muss ich das noch veröffentlichen, das Game. Sonst könnt ihr es nicht spielen. Vielleicht noch vorher, bevor wir jetzt das machen, wo ich das hier gerade sehe mit dem Semesterprojekt. Ich habe das jetzt mal veröffentlicht hier. Kommt noch in meine E-Mail rein, die ich euch nachher schreibe.

Und es wird noch eine Ergänzung geben müssen. Ich habe doch noch von der Raumverwaltung nicht die Antwort, aber entweder Manuel oder ich werden in einen anderen Raum ausweichen. Also einer von uns wird hier sein. Dann werde ich dann jeweils bei den Gruppen noch, also von dem anderen, dann den Raum dazu schreiben. Schreibe ich euch dann nochmal. Aber das hattet ihr schon mal gesehen, oder? Ich habe es jetzt auch auf dem Moodle. Coaching Slots oder sowas heißt es für den zweiten, vierten.

Genau, also jede Gruppe hat eine Dreiviertelstunde. Ihr seht, wer euer Coach ist und dann kommt noch der Raum. Habt ihr noch irgendwelche Fragen zum Semesterprojekt jetzt gerade?

Organisatorisch, inhaltlich können wir auch nächste Woche klären. Ja? Ja, so gerne. Für unsere Gruppe werden Berufe, Rennspiele, Medizin und so weiter verheißen.

Staud, deswegen wollte ich lieber vierer Gruppen. Könnt ihr nochmal gucken auf Moodle, ob ihr eine andere Zweiergruppe findet? Ich hatte das Gefühl, es gibt Zweiergruppen. Oder ihr verteilt euch auf zwei Gruppen. Mir ist das recht, also ich habe nichts dagegen. Also man sieht es ja hier. Also hier jetzt...

Am Mittwoch gibt es keine Zweiergruppe. Inzwischen sind alle zu dritt. Ja, ich würde auch dem zustimmen. Es ist zu zweit schon herausfordernd. Und für uns ist es auch angenehmer, wenn ihr euch dann aufteilt. Kriegt ihr hin, oder?

Also die Gedanken, die ihr euch jetzt zu zweit gemacht habt, könnt ihr dann einbringen. Genau. Super. Schafft ihr. Gut, jetzt mache ich das hier mal verfügbar. Also jetzt findet ihr das unter den Folien im Link.

Das Prinzip kennt ihr ja schon und ich hoffe, dass es funktioniert. Das habe ich ganz neu gebaut. Ja, falls irgendwas nicht so rauskommt, wie es soll, werden wir eine Lösung finden. Hauptsache ihr versteht, wie das Oversampling funktioniert. Also, ich warte darauf, die Gruppen zu sehen, wie sich ihre Köpfe zusammenstecken.

Gruppen, genau, Gruppen. Okay. Also, ich würde noch einmal durchgehen, damit wir noch mal, ich mache alles von vorne bis hinten, damit wir noch mal für uns nachvollziehen, was jetzt eigentlich passiert ist. Ist ja bei so einem Game immer wichtig, man taucht dann so ein und dann ist man

draußen und wenn man sich dann fragt, was habe ich eigentlich gemacht, weiß man es manchmal nicht mehr so genau.

Also gut, erstmal musste man hier diese Dose öffnen und dann die Frage. Also es ist ja nicht genau das, was auf der Folie stand. Auf der Folie hatte ich leicht andere Maße, deswegen musste man nochmal kurz nachdenken, was man der Folie entnehmen konnte. Also ich nehme meine konkret jetzt die hier, dass die Genauigkeit sinkt. Insbesondere passiert es deswegen, weil durch Overtampling typischerweise

eine ganze Menge False Positives produziert werden. Also es wird optimistischer und dadurch entstehen eben viele False Positives. Aber die Kosten sinken. Das ist sozusagen ja auch einer der Gründe, warum wir das Ganze machen. Und im Game war jetzt nach Precision und Recall gefragt, nicht nach AUC und F1. Ja, Precision sinkt natürlich auch aufgrund der vielen False Positives. Also ich habe dann

Eine geringere Trefferquote rufe öfter an, ohne dass jemand Interesse hat. Aber ich finde eben auch mehr von den Nadeln. Also der Recall steigt. Das heißt, wenn man jetzt diese vier Sachen anschaut, Accuracy sinkt, Precision sinkt, Kosten sinken, nur der Recall steigt. Also musst du hier eingeben 0010 bei der Spüle. Und dann, das war jetzt mal die generelle Überlegung, dann holt man sich hier den Schlüssel aus dem Papierkorb und kann

den Kleber und das Papier da holen und vor allem auch die Anweisung, was als nächstes kommt. Ja, also die Kostenmatrix, die habt ihr alle hingekriegt, oder? Hier steht ja, es soll nur in einer Spalte was stehen, was nicht Null ist. Jetzt ist die Frage, wollen wir die Kostenmatrix gleich so hinschreiben, wie die Confusion-Matrix später in Orange ausgegeben wird? Vielleicht keine schlechte Idee. Also die Kostenmatrix...

dann auch in dem Design, dass wir erst No und dann Yes haben, würde nie aussehen. Also wenn Yes, dann rufen wir an. Und das Szenario sagt, dass es im Schnitt sechs Minuten dauert, so ein Gespräch, und dass die Stunde 150 Euro kostet. Das heißt, sechs Minuten zufälligerweise sind eine zehnte Stunde, also kostet es

15 Euro für ein false positive. Also wenn das Modell No sagt, dann machen wir nichts, dann haben wir null Post. Und dann, das war vielleicht noch ein bisschen tricky, beim True Positive, wenn ich also anrufe, kostet wieder 15 Euro, aber dafür investiert jemand dann im Schnitt 10.000 Euro und 100 Euro Gewinn, also 1% mache ich.

leider abzüglich der 15 für das Gespräch, also Gewinn von 500. Soweit klar. Da seid ihr ja alle hingekommen, oder? An diese Kostenmatrix. Das war jetzt wirklich wieder so eine Trockenübung, so wie bei unseren zwei Fällen letzte Woche. Und dann musste man multiplizieren, also da am Ende das Vorzeichen weggelassen wird, kann ich es wieder umdrehen. Also

Ich habe 10 false positives und 19 true positives. Also ich habe sozusagen 85 mal 19 Gewinn und von dem muss ich dann wiederum 15 mal 10, also 150 abziehen. Das war auch klar soweit, oder? Also muss ich dem Drucker Papier geben und dann gebe ich das ein und dann sagt er mir das mit dem Oversampling. So, jetzt können wir natürlich hier das alles eingeben, 72 und 42 und dann kriegen wir die Hinweise.

Die müssen wir jetzt nicht alles durchlesen, sondern ich würde es direkt einfach mal machen. Und vielleicht nochmal jeweils zurück zur Folie springen, um zu sehen, was wir da eigentlich tun. Also die Idee war ja, und wir wollen Oversampling machen, also das rechts. Was ist die Idee? Die Idee ist, wir

nehmen Daten, und zwar die vollen Daten, und erstmal, das ist das, was wir jetzt in Orange tun werden, machen wir sozusagen eine Kopie.

Also am Ende wird es so aussehen. Das, was ihr gebaut habt, um zu entkommen aus dem Raum, sah so aus. Also nicht ganz so viel Oversampling, wie ich auf der Folie ursprünglich gemacht hatte. Etwas weniger, aber immerhin die Nadeln verdoppelt. Und ursprünglich, also machen wir mal erst das hier. Nur das. Wie geht das?

Was der Workflow macht, den ich euch zur Verfügung gestellt habe, ist hier 20% der Daten wegzunehmen und diese Daten fließen hier rüber und werden als Testdaten verwendet. Also diese, nee, es stimmt gar nicht 20%, oder? Hier 1000. Also wenn ihr das aufsummiert, kommt immer 1000 raus. Also 1000 Kunden werden als Testdaten hier rüber geschickt. Und ihr seht, zum Trainieren,

Und das ist wichtig hier, das ist ja ein bisschen klein geschrieben, aber wichtig. Also die 1000 gehen hier rüber und alle anderen, da steht hier Remaining Data, also ungefähr 3600 oder so, werden zum Trainieren verwendet. Also 3600 Trainings und 1000 Tests, irgend sowas in der Größenordnung. Und wenn ich jetzt erstmal das hier machen will, also eine Kopie, dann mache ich hier eine Verbindung und

Wir werden gleich verstehen, warum Concatenate. Und da ist es jetzt wichtig, wieder diese 3600 zu nehmen, die ich auch da oben zum Trainieren verwendet habe. Jetzt habe ich also einfach eine Kopie gemacht. Hier raus. Und jetzt möchte ich einmal die Nadeln da oben drauf stapeln. Okay? Nochmal. Das war das Trill. Und das mache ich, indem ich mir nochmal diese Daten nehme. Auch hier muss ich wieder doppelklicken und die 3600

rauspicken, also im Prinzip nehme ich nochmal alles und jetzt mache ich die blauen weg. Dann habe ich noch einen roten Block, den ich da hier so drauf staple. Und das mache ich hier, indem ich sage, Y ist Yes als Bedingung. Das selektiert mir die interessierten Kunden, also die roten Nadeln. Und die klatsche ich dann zusammen. Ihr seht, hier gibt es noch Primary Additional Data, das klappt so. Und jetzt habe ich

die Daten, so wie auf der rechten Seite. Und mit denen trainiere ich jetzt wieder mein Modell. Und dann sagte das Game hier noch, Rich mit C gleich 1 passt. Und dann wiederum wichtig, evaluieren tue ich auf den gleichen 1000 Kunden wie vorher auch. Und kann jetzt vergleichen, ob mein Modell optimistischer geworden ist. Also das werden wir gleich sehen, dass jetzt deutlich mehr Yes-Vorhersagen stattfinden.

Also vorher waren es 29, jetzt sind es 76. Natürlich auch deutlich mehr. Fantastisch, jetzt kommen nicht die Zahlen raus, die ich euch gegeben hatte, sondern wieder andere. Also ich muss wahrscheinlich dieses Game doch irgendwie anders designen. Das ist einfach nicht stabil. Aber die Zahlen sind schon ungefähr immer in dem gleichen Range. Das heißt, es wird mutiger.

Und dadurch haben wir auch höheren Gewinn. Also wenn ihr die Kosten neu ausrechnet. Ich schreibe jetzt die nochmal an Schafe, die wir im Games verwenden mussten. Also da waren es 854 True Negatives, 40 False Positives, 38 True Positives, 68 False Negatives. Okay. Und dann wieder die gleiche Rechnung.

Das müssen wir jetzt nicht unbedingt nochmal rechnen. Doch? Nein? Also was rechnen wir? Wir machen es nicht, aber wir wissen, wir müssen minus 85 mal 38 rechnen und dann plus 15 mal 40 und da kommt dann minus 2630 raus. Richtig? Und da wir das Vorzeichen weglassen sollen, sieht das dann so aus. Und dann klappt es auch. Und da jetzt schrauben hier, dann kann ich es abschrauben.

Und hier fragt mich das Spiel dann noch nach der Accuracy. Also die könnt ihr jetzt, wenn ihr wisst, dass die Testmenge aus tausend Kunden besteht, eigentlich schon fast im Kopf ausrechnen. Also ihr addiert einfach die richtigen Vorhersagen zusammen. Hier sind das. Was kommt da raus? Oh, ja. Irgendwas stimmt doch da nicht, oder? Ja.

Also wir müssen es doch ausrechnen hier. Also 90,3 wäre hier die Accuracy. Und hier war es, glaube ich, 89,2, oder? Ja, okay, ich bin nicht so gut im Kopf. Und dann ist die Differenz zwischen den beiden 1,1 mal 200, soll man dann noch rechnen, steht da, multipliziert mit 200, also 220 und dann Escape.

Noch Fragen, bevor wir heimgehen und nächstes Mal uns, das will ich natürlich auch noch hier eskalieren, zum Coaching treffen. Also Fragen zum Thema Oversampling. Also es tut mir leid, es war ein experimentelles Game. Ich hoffe, es hat trotzdem Spaß gemacht.

Wenn keine Fragen mehr sind, ja, wie gesagt, es gibt wieder ein Quiz und natürlich braucht ihr noch die Raumangaben. Ich schreibe eine E-Mail. Und dann sehen wir uns nächste Woche zum Coaching. Ah, habt ihr es gesehen? Wahrscheinlich nicht. Ich schaue hier auch nicht dauernd hin. Es sieht so aus jetzt. Der Manuel wird übernehmen dann nach dem Coaching für diese drei Unterrichtseinheiten.

Und dann bin ich am Ende nochmal da. Aber dann, also im Unterricht sehen wir uns jetzt erstmal nicht mehr. Endspurt. Dann wieder zum Endspurt, ja. Coaching for Art, ja. Also ich meine, hier ist ja nochmal eine Brücke. Also der Freitag ist die Brücke, deswegen machen wir den Mittwoch auch nicht. Ja.

Also der 1. Mai. Freitag dann. Sonst noch Fragen?