STØTTET AF
mælkeafgiftsfonden

Projekt 5517: MAF Datadrevet management i mælkeproduktionen – AP2
Arbejdsgruppe: Risiko for udsætning

Afrapportering 2021 – Risiko for udsætning

I regi af AP2 i Projekt 5517: MAF Datadrevet management i mælkeproduktionen blev der nedsat en arbejdsgruppe, som skulle kigge på risiko for udsætning og muligheder for at udvikle modeller, som kan udpege kvier og køer, som er i risiko for udsætning.

Gruppen blev i første omgang dannet af fire medarbejdere. I efteråret blev tilknyttet en femte medarbejder. Dermed havde gruppen kompetencer inden for kvægbrug, statistik og data science.

Ud over projektmøder blev der i arbejdsgruppen afholdt to møder før sommer og to møder efter sommer.

Indledende overvejelser

I møderne inden sommer tog vi udgangspunkt i punkter, som var givet til arbejdsgruppen:

Områder	**Anvendelse og grundlag**
Risiko for udsætning i 1 laktation	Udvikling af en model, som kan udpege kvier, som har en øget risiko for ikke at nå 1 kælvning eller ikke kommer gennem 1 laktation.
Risiko for udsætning i næste laktation	Udvikling af en model, som kan kvantificere risikoen for at en ko bliver udsat inden afslutning på næste laktation.

Her blev bl.a. diskuteret forskel mellem *udskiftning* og *udsætning*, og det blev diskuteret hvilken gavn en udpegning af risikodyr kunne gøre hos landmanden.

Uden at være helt præcis kan udskiftning være tilfælde, hvor en ko sættes ud, fordi der er en kvie, som skal ind i malkestalden. Og udsætning kan være tilfælde, hvor landmanden er nødsaget til at sende koen til slagtning (eller koen dør eller skal aflives.)

Ift. gavn hos landmanden blev der diskuteret muligheder for at fede en ko op til slagtning i stedet for at anvende den til mælkeproduktion, eller muligheder for særlig pleje og pasning ved forhøjet risiko for udsætning, fx hjælp til kælvning og/eller indmalkning. Desuden også optimale tidspunkter for udsætning – at forskellige landmænd kan have forskellige strategier.

Det blev også diskuteret, hvornår en alarm skulle gives til landmanden. Fx lige før eller lige efter en kælvning.

Det blev i løbet af arbejdsgruppemøderne bestemt at fokusere på ufrivillig udsætning, og ikke på udsætning af køer som kunne være en form for ønsket udskiftning.

Dette arbejde resulterede i tre mulige fokusvinkler / opgaver:

1. Risiko for udsætning i start af laktation
2. Risiko for udsætning i løbet af laktationen – eller koen levetid
3. Besætningsanalyse

For alle tre opgaver skal der arbejdes med følgende spørgsmål:

a) Hvad kan landmanden bruge det til?
b) Hvornår skal landmanden have information eller alarm?
c) Hvilke indsatspunkter eller handlinger kan anbefales til landmanden?

De tre mulige opgaver blev fremlagt ved projektmøde den 28-06-2021:

Opgave 1
Alarm ved risiko for udsætning i start af laktation

- Risiko: uønsket udsætning (død, aflivning, slagtning) inden for 60 dage efter kælvning
- Alarmtidspunkt: før (eller umiddelbart efter) kælvning
- Risikofaktorer:
  - kælvningsnr, race, kælvningsinterval, kælvningsalder, sædtype, avlsværdier (både koen og kalven), vægt, (kælvningsbesvær/dødfødsel,) årstid, besætning/besætningstype
  - goldperiodelængde, goldydelse, celletal ved goldning, behandling i goldperioden, sygdomme i starten af sidste laktation, huld ved goldning, fedtsyresammensætning i slutning af laktation, fedt/protein-forhold, ydelse i det hele taget
- Værdi: landmanden kan tage action allerede i forbindelse med at koen skal kælve eller kort derefter.

Prioritet: 1-2

Opgave 2:
Alarm ved risiko for udsætning i løbet af laktationen

- Risiko: for udsætning i løbet af laktationen (eller beregning af mest sandsynlige levetid)
- Her nærmer vi os dog at kunne udpege, hvilke dyr, som det for landmanden er mest optimalt at udsætte. Dette er en større opgave, og det arbejdes der med i andre projekter

Prioritet: 3

Opgave 3, besætningsanalyse:
Hvilke faktorer betyder mest for udsætning?

- Formål: at anvise faktorer, som der kan arbejdes med i den enkelte besætning, for at der bliver udsat færre køer.
- Fokus på forskellige udsætningsgrupper, rangeret efter ufrivillighed: 0-60 dage (efter kælvning), 61-180 dage højt ydende, 61-180 dage lavt ydende, > 181 dage drægtige, > 181 dage ikke drægtige
- Risikofaktorer:
  mastitis, mælkefeber, kælvningsbesvær og tvillingefødsel, sædtype, tilbageholdt

efterbyrd, børbetændelse, løbedrejning, ketose, halthed, manglende drægtighed, afstand fra kælvning til inseminering, lavt huld, lav ydelse, eksteriørmål, varmestress ved kælvning, avlsværdi, paratb-prøver, flytninger, årstid
- Værdi: landmanden vil kunne få anvist faktorer, som der kan arbejdes med, for at der bliver udsat færre køer.

Prioritet: 1-2

Anvendelse af Azure ML til at beregne alarm ved risiko for udsætning i start af laktation

I efteråret opstod der mulighed for at afprøve Azure ML (Microsoft-bud på Machine Learning). I den forbindelse valgte vi at kigge på opgave 1 ovenfor: Alarm ved risiko for udsætning i start af laktation. Til dette arbejde blev anvendt et datasæt, som har en stor del af de informationer, som er nævnt som risikofaktorer i opgaven.

I det følgende beskrives derfor:
- I ultrakort form: Azure ML
- Det anvendte datasæt
- Overvejelser omkring præcision af alarm
- De opnåede beregninger fra Azure ML

Til sidst i afsnittet vil jeg samle op på, hvor disse resultater kan bringe os hen.

Om Azure ML – i ultrakort form

Azure ML er Microsofts bud på en platform til Machine Learning (ML) i skyen. Man kan læse mere om det på følgende links:

- Azure ML AutoML experiment: https://bit.ly/3l8UlIA
- Azure ML Pipeline using the Designer: https://bit.ly/3A5eQdt
- Azure ML Notebook experiment: https://bit.ly/3B9vXwi

Additional material

- Azure ML and R:
  - https://azure.github.io/azureml-sdk-for-r/index.html
  - https://medium.com/microsoftazure/azure-machine-learning-for-r-practitioners-with-the-r-sdk-323454d338ae
- Azure ML GitHub:
  - https://github.com/Azure/azureml-examples
  - https://github.com/Azure/azureml-previews

Bl.a. vælger man et compute cluster og kan importere datasæt. Derefter kan man sætte en Automated ML i gang. Dvs. at Azure ML afprøver en række ML-modeller på det datasæt, som man har indlæst og specificeret. Der er også mulighed for at "sætte modellen i produktion" ved direkte at anvende en Deploy-funktion, men dette blev ikke brugt i denne gruppes arbejde.

Alle ML-modeller er specificeret som en Python-pakke-version.

Alt i alt fremstår Azure ML som brugervenligt med mange muligheder for "peg og klik", men når man kommer til nogen mere specifikke og detaljerede ønsker, syntes det ikke altid at være muligt at håndtere. Dog fik vi i projektet afprøvet en række ML-modeller på en del forskellige datasæt – alle værende en del af det oprindelige datasæt.

Der er screen dumps fra Azure ML i de to bilag sidste i dette notat.

Det anvendte datasæt

Det datasæt, som er anvendt i Azure ML, bygger på et tidligere dataudtræk, som oprindeligt er udtrukket for en 20 års periode: januar 2000 – slutningen af 2020. Selv udtrækket fra Kvægdatabasen tog den gang 10 timer. Dertil kom databearbejdning og sammenfletning. Dette gav et datasæt med i alt ca. 15 millioner datalinjer, som hver repræsenteret én kælvning fra køer, som har ydelseskontrolleringer.

Datasættet, som blev anvendt til Azure ML, kommer fra kælvninger fra ydelseskontrollerede køer fra 2015 til 2020. Da ikke alle features findes for 1. kalvs køer - og da 1. kalvs køer har klart mindre risiko for død/udsætning – blev det valgt kun at koncentrere sig om 2. og senere kælvninger. Desuden blev der renset op i data på forskellig vis. Bl.a. blev der sørget for, at der ikke var missing values i datasættet, der blev sat grænser for en række features for at undgå meget specielle dataværdier. Fx krævedes at drægtighedslængden var mellem 278-22 og 282+22 dage, og alder ved første kælvning mellem 18 og 40 måneder. Også tvillinger blev frasorteret - for dem ved vi under alle omstændigheder medfører større risiko for koen, og fordi vi så ikke har entydig feature på kalvens køn.

Dermed blev der til ca. 1,5 millioner kælvninger (datarækker) i perioden 1. juli 2015 til og med 30. juni 2020, som indgik i afprøvningen med Azure ML.

Datasættet har følgende indhold:

Features		Kælvningsnr.
		Koens race
		Koens alder v/ kælvning
	IKKE ved 1. kælvning (dublet-værdi)	Kælvningsinterval
		Alder v. 1. kælvning
		Koens tilstand ved fødsel
		Antal insemineringer
		Sædtype (køns-sorteret)
		Tyrens race
		Tyrens race - Alternativ 1
		Kælvningsmåned
	Relateret til kælvningen	Kode for kælvningsforløb
		Kalvens tilstand ved fødsel
		Kalvens tilstand v. fødsel - Alternativ 1
		Kalvens køn
		Drægtighedslængde
	IKKE ved 1. kælvning	Ydelse ved goldning
		Celletal ved goldning
		Fedt/protein start lakt.
		Fedt/protein slut lakt.
		Gennemsnitlig ydelse
Target	Målt inden for 60 dage efter kælvning	Død (Død+aflivet)
		Slagtet
		Udsat (død eller slagtet)

Dermed har det nogle features/risikofaktorer, som ikke er beskrevet ovenfor i ”opgave 1”, nemlig:

- Alder ved 1. kælvning

- Koens tilstand ved dennes fødsel
- Antal insemineringer
- (Tyrens race)
- Kalvens køn
- Drægtighedslængde

Og følgende features/risikofaktorer fra "opgave 1" ovenfor var *ikke* med i datasættet:

- Avlsværdier (både koen og kalven)
- Vægt
- Besætning / besætningstype
- Goldperiode
- Behandling i goldperioden
- Sygdomme i start af sidste laktation
- Huld ved goldning
- Fedtsyresammensætning i slutning af laktation

I ovenstående skema bruges ordet "Features", fordi det er den term, som anvendes i ML, men de kan også kaldes risikofaktorer. I ML anvendes termen "Target" om den værdi/hændelse, som man ønsker at forudsige. I nogle type af analyse kaldes en sådan variabel også 'respons'.

Som target lagde vi op til tre muligheder:

- Død (både død og aflivning)
- Slagtning
- Eller begge, kaldet udsætning: dvs. at både død og slagtning var en hændelse

De features som er relateret til kælvningen kunne kun bruges i modellerne, såfremt vi regnede med en alarm *efter* kælvning.

Overvejelser omkring præcision af alarm

Inden afprøvningerne i Azure ML havde vi en drøftelse af, hvor godt en model skal performe, for, at den er relevant at kunne bruge i praksis. Det findes der ikke nødvendigvis et entydigt svar på, men vi kom frem til nogle kriterier.

Præcisionen af en alarm beskrives bedst ved en såkaldt "confusion matrix", eller direkte oversat til dansk: forvirringsmatrice. På sin vis et lidt mærkeligt ord, når formålet med matricen gerne skulle være at skabe et overblik. Men en confusion matrix kan se således ud:

		Sand tilstand		I alt
		Død	Levende	
Forudsagt tilstand	Død	13	27	40
	Levende	7	953	960
I alt		20	980	1.000

Her er valgt en situation, hvor vi har 1.000 kælvninger, og 20 af disse kælvninger resulterer i at koen dør inden for 60 dage efter kælvning. Ved denne fiktive forudsigelse/alarm, vil der blive udpeget 13 af disse døde køer på forhånd, mens der bliver udpeget 27 alarmkælvninger, som dog ikke resulterer i en død ko inden for 60 dages.

De 7 kælvninger med den rødlige baggrund er falsk negative, mens de 27 med den rødlige baggrund er falsk positive. Vi ønsker selvfølgelig at minimere disse antal af falske (forkerte) udmeldinger.

Generelt kan en confusion matrix beskrives som:

		Sand tilstand		I alt
		Død	Levende	
Forudsagt tilstand	Død	TP	FP	TP+FP
	Levende	FN	TN	FN+TN
I alt		TP+FN	FP+TN	Total N

Hvor

- TP er True Positive, sandt positive
- FP er False Positive, falsk positive
- FN er False Negative, fald negative
- TN er True Negative, sandt negative

Til denne confusion matrix tilknyttes ofte en række beregninger – med forskellige mål for præcision eller nøjagtighed. I det følgende holder vi os til to mål, nemlig:

- Sensitivitet = TP / TP+FN – altså kvotient fra første *kolonne*
- Positive Predictive Value (PPV) = TP / TP+FP – altså kvotient fra første *række*

Sensitivitet fortæller, hvor god modellen er til at finde de positive tilfælde (altså døde køer). PPV fortæller, hvor mange reelle risiko-køer, som vi vil finde blandt de køer, som bliver udpeget som værende alarm-køer.

På mødet i arbejdsgruppen kom vi frem til, at et godt bud er, at vi skal finde mindst halvdelen og gerne 2/3-dele af risiko-køerne. Dvs. en sensitivitet på mindst 50% og gerne på 60-70%.

Vi skal give så gode alarmer, at alarmer på fx 3 køer med alarm peger på 1 ko, som reelt er en risiko-ko. Dette svarer til en PPV på 33%. Hvis det drejer sig om en alarm *før* kælvning, er det ok med en dårligere alarm. Fx ok, med at kun en 1/10 af alarmer rammer rigtigt i stedet for 1/3. Altså ok med PPV på 10%.

Vi forventer, at selve kælvningen er en stor risikofaktor i sig selv for det videre forløb for koen - altså kælvning, og hvad der evt. følger af sygdomme i de første 3, 4 til 7 dage efter kælvning. Det kunne derfor måske være en god idé at udpege risiko for "kælvningsproblemer". Det kan være nemmere for landmanden at gøre noget ved - fx øget fokus, overvågning og hjælp ved kælvning. Og en afledt effekt af det vil så give lave udsætning efterfølgende.

Ovennævnte tal om præcisionen af en alarm er kun et første bud. De vil skulle afprøves i en reel situation, hvor landmænd kan afprøve og vurdere alarmerne.

I ovenstående confusion matrix med tal-eksemplerne, er

- Sensitiviteten = 13 ud af 20 = 65%, og
- PPV = 13 ud af 40 = 33%

Det er altså eksempel på en confusion matrix, som illustrerer det, som vi forventer at være en acceptabel god model for alarm.

De opnåede beregninger fra Azure ML

Vi afviklede ca. 9 runs i Azure ML i alt. Ét run er at Azure ML kører måske 20 eller 30 ML-modeller igennem, og derefter viser forskellige vurderingskriterier for hver model. Hvert run kører altså på hver sine egne data. Dog er alle disse ni datasæt med udgangspunkt i det ovenfor beskrevne datasæt. De ni runs kan opdeles i to forskellige grupper, som hver fokuserer på forskellige måder at inddele data:

- I første gruppe af runs arbejder vi med forskellige muligheder for up- og down-sampling. Dette for at rette op på, at data er ret skævt på den måde, at der kun er ca. 2% af køerne, som dør. Mange standard ML-metoder fungerer mere optimalt, hvis det inddelingen var mere fifty-fifty – altså at ca. 50% af køerne døde (er risikokøer).
- Den anden gruppe af runs fokuserer på forskellige targets og forskellige alarmtidspunkter. Hvor første gruppe af runs udelukkende fokuserer på koens *død* og med et alarmtidspunkt *efter* kælvning, fokuserer disse runs på at alarmen kan gælde både koens død, men også udsætning generelt, og det kan kombineres med et valg om alarmen skal være før eller efter kælvning.

Der hvor potentialet synes at være størst, er i den anden gruppe af runs. Der gives dog en gennemgang af begge grupper af runs.

Men inden en gennemgang de to grupper af runs, beskrives her meget kort, hvilke vurderingskriterier, som er blevet anvendt i Azure ML. Dette også for at vise, hvordan Azure ML viser forskellige slags score på modellerne.

Vurderingskriterier	Meget kort beskrivelse	Bemærkning
AUC weighted	Areal under ROC-kurven. Tallet er som regel over 0,5 og kan maks. være 1. Jo højre, jo bedre	Kan blive kunstigt høj, hvis andelen af risikohændelser er meget lav som i vores tilfælde.
Norm macro recall	Ligger mellem 0 og 1. Jo højre, jo bedre. Hænger sikkert sammen med recall, som er det samme som sensitivitet.	
F1 score macro	Ligger mellem 0 og 1. Jo højre, jo bedre. Hænger sikkert sammen F1 score, som er det harmoniske gennemsnit af 'precision' og 'recall'. Dvs. disse størrelse kan beregnes ud fra confusion matrix.	
ROC	ROC er en kurve, hvor man tegner sensitivitet (recall) op mod (1 – specificitet), hvor variationen over kurven udgøres af det cut-off, som man frit kan vælge. Men hvis man øger sensitiviteten, vil det ske på bekostning af specificitet og vice versa.	ROC-kurver viser dermed med resultater for flere cut-offs. Recall og F1 score synes i modsætning hertil, kun at være beregnet for en 'middel-cut-off' (50%).

Første gruppe af runs, hvor vi afprøver forskellige slags up- og down-samplinger

I dette datasæt arbejder vi med, at Target er koens død inden for 60 dage efter kælvning, og vi arbejder med, at alarmen for risiko for død skal gives umiddelbart *efter*, at koen har kælvet.

Der er ca. 2-3% af køerne som dør, dvs. at disse datasæt er meget skævt. Det kan derfor nogle gange være en god idé at

- Up-sample, dvs. at replikere de datalinjer, som indeholder info om køer, som dør, for dermed at komme tættere på en fifty-fifty-situation.
- Down-sample, dvs. at vi kun udvælger en del af de køer, som ikke dør. På den måde bliver der mere ligelig fordeling mellem datalinjer med køer, som hhv. dør og ikke dør.

Til de fem runs arbejdede vi med fire datasæt med forskellige slags up- og down-sampling:

Datasæt	**Beskrivelse**	**Noter**

stet2Plus_FiveYear	Det fulde datasæt til Azure ML	Ca. 1,5 millioner kælvninger i perioden 1. juli 2015 til og med 30. juni 2020. For 2. og senere kælvninger hos ydelseskontrollerede køer. Alle feature er uden missing values. Ca. 2,5% af køerne er døde inden for 60 dage.
stet2Plus_FiveYear_Up10	Med udgangspunkt i det fulde datasæt har vi up-samplet de døde køer med en faktor 10	Ca. 1,8 millioner kælvninger/datalinjer, hvor døde køer er up-samplet, sådan at der her er ca. 20% døde køer.
stet2Plus_FiveYear_DN39	Med udgangspunkt i det fulde datasæt har vi down-samplet de levende køer	Ca. 75.000 kælvning, hvor levende køer er down-samplet, så der er præcis lige så mange levende køer som døde køer. Dvs. nu 50% døde køer, og dermed et præcist balanceret datasæt.
stet2plus_5year_updn	Med udgangspunkt i det fulde datasæt, som er up- og down-sampled med hhv. 2x og 10% i Azure ML Pipeline	Dermed er der ca. 1/3 døde køer i datasættet.

Da vi lavede et run på første af ovenstående datasæt, havde vi valgt standardindstillingen: at optimere efter AUC weighted. Det gav dog det problem at recall blev beregnet til nul. Det svarer lidt til en helt simpel situation, som illustreret ved følgende confusion matrix:

		Sand tilstand		I alt
		Død	Levende	
Forudsagt tilstand	Død	0	0	0
	Levende	20	980	1.000
I alt		20	980	1.000

… altså at modellen blot forventer at alle køer overlever, og dermed gætter modellen jo kun forkert i ca. 2% af tilfældene.

Derfor valgte vi i de senere runs af optimere efter Norm macro recall i stedet for AUC weighted. I skemaet nedenfor ses, at der er to runs på samme datasæt, nemlig run nr. 1 og run nr. 66. Ved run nr. 1 blev der optimeret efter AUC weighted, mens der ved run nr. 66 blev optimeret efter Norm macro recall. Det gav så i øvrigt ca. samme AUC weighted ved de to runs.

Run nr.	Datasæt	AUC weighted	Norm macro recall	F1 score macro	Antal modeller med generelt topniveau
1	stet2Plus_FiveYear	*0,73	0,000	0,49	Mange
66	stet2Plus_FiveYear	0,72	*0,329	0,44	Mange
166	stet2Plus_FiveYear_Up10	1,00	*0,999	0,99	*Kun én*
285	stet2plus_5year_updn	0,97	*0,874	0,94	*Kun én*
340	stet2Plus_FiveYear_DN39	*0,73	0,340	0,67	Mange

* Angiver om den angivne Run er optimeret efter AUC eller Recall. Dvs. hvis AUC er angivet, så har Azure ML anvendt en række modeller og testet, hvilken model, som gave den bedste AUC. Og for nogle Runs har Azure ML altså søgt efter bedste Recall.

Man kan også bemærke, at ved de to runs med up-sampled datasæt (nr. 166 og 285) får vi en 'fantastisk' høj AUC weighted – næsten på 1. Men vores vurdering er, at det skyldes at datasættet netop er up-sampled, dvs. at nøjagtig samme datalinje med en død ko ses flere gange i datasættet, og derfor har modellen nemt ved at genkende, at der er tale om samme linje, og dermed samme udkomme (dvs. target, altså at koen dør). Det ses også, at ved disse to runs er det kun én af de afprøvede modeller, som giver et sådan 'fantastisk' resultat. Hvis vi kigger på næstbedste model, får vi et mere realistisk billede:

Run nr.	AUC weighted	Norm macro recall	F1 score macro
166	0,73	0,34	0,61
285	0,72	0,33	0,64

At det sidste run (nr. 340) er optimeret efter AUC weighted, gør ikke så meget, fordi der her er tale om et pænt balanceret datasæt (præcis 50% af køerne dør).

Der kan generelt være en række problemer ved at up-sample og down-sample ens datasæt. Nogle af disse problemer kan man arbejde sig uden om eller korrigere for, men det fandt vi ikke umiddelbart muligt i Azure ML.

Men når vi ser på hvordan de ovenstående runs performer, så er der måske heller ikke den helt store forskel. Særligt ikke, når vi 'nøjes' med at kigge på næstbedste model i de ro runs nr. 166 og 285. Så har vi generelt en AUC weighted på omkring 0,73 og Norm macro recall på 0,33.

De tal, som er angivet i de to ovenstående tabeller, kommer fra output fra Azure ML. I bilag 1 til dette notat ses screen dump af disse output.

På grund af de ovennævnte problemer med up- og down-sampling, og fordi de forskellige datasæt syntes at give næsten samme resultat, blev det valgt ikke at arbejde mere med disse teknikker.

Til gengæld blev der foretaget nye runs på andre datasæt med forskellige targets og forskellige alarmtidspunkter. Det beskrives i næste afsnit.

Anden gruppe af runs : med forskellige targets og forskellige alarmtidspunkter

I nedenstående bilag 1 for runs med et datasæt, som fokuserer på target = død og alarm efter kælvning, ses også hvordan de forskellige features bedømmes ift. importance, dvs. en form for mål for, hvor meget de betyder for beregningen af alarmen.

Det er fx koens alder og goldningsydelse, som vægtes højest. Først på tredjepladsen eller senere ses features, som er tilknyttet selve kælvningen, fx forløbskode. Der er grænser for, hvor meget man kan tolke på dette, men det kunne antyde, at data fra kælvningen ikke er så voldsomt vigtige for beregningen af alarmen. Og da det fra den kvægfaglige side antydes, at præcisionen af alarmen (specifikt: PPV) ikke behøver at være så stor for alarmer *før* kælvning, så giver det et lille håb om, at vi måske kan komme bedre i mål med en god alarm, hvis vi kigger på alarmer før kælvning.

Derfor beskrives her også vores arbejde med fire scenarier, hvor vi både ændre på target og alarmtidspunkt.

Runs i Azure ML med disse 2 x 2 = 4 scenarier gav følgende resultater:

Nr.	Datasæt	AUC weighted	Norm macro recall	F1 score macro	False vs. True Positive Rate (x,y) i ROC	Prevalens (andelen af risikokøer)
A	Død, Alarmtid: efter kælvning	0,72	*0,33	0,44	(10% , 33%)	2.524943
B	Død, Alarmtid: før kælvning	0,70	*0,32	0,43	(10% , 30%)	2.550654
C	Udsat, Alarmtid: efter kælvning	0,70	*0,30	0,47	(10% , 29%)	5.163436
D	Udsat, Alarmtid: før kælvning	0,67	*0,26	0,44	(10% , 25%)	5.261305

'Død' relaterer til at koen er død eller aflivet inden for 60 dage efter kælvning.
'Udsat' relaterer til at koen er død, aflivet eller sendt til slagtning inden for 60 dage efter kælvning.

Som i afsnittet ovenfor kommer tallene i ovenstående tabel fra output fra Azure ML. Dette output ses i bilag 2 som screen dumps.

Det ses, at der sker en svag gradvis forringelse af ML-modellerne fra A til D, når vi måler på AUC weighted og Norm macro recall.

Men man kan også vælge at se nærmere på de punkter på ROC, som er aflæst visuelt, fx (10% , 30%). Da vi dermed kender både sensitivitet, specificitet og prævalens, kan vi udregne, hvordan den tilsvarende confusion matrix ser ud, og dermed også beregne PPV. For det var fra starten sensitivitet og PPV, som vi brugte som kriterier til at vurdere, hvor god en alarm var.

De fire scenariers confusion matrices ses i nedenstående skemaer. Sensitivitet (se) og PPV er beregnet i næstnederste hhv. tredjenederste linje:

Scenarie A (1)		Sand tilstand		I alt
Død, Alarmtid: efter kælvning		Død	Leve	
Forudsagt tilstand	Død	8	97	106
	Leve	17	877	894
I alt		25	975	1.000
False Positive Rate		10%		PPV 7,9%
True Positive Rate	(Se)	33%		
Prevalence		2,525		

Næsten samme tal, men nu alarm *før* kælvning

Scenarie B (2)		Sand tilstand		I alt
Død, Alarmtid: før kælvning		Død	Leve	
Forudsagt tilstand	Død	8	97	105
	Leve	18	877	895

I alt		26	974	1.000
False Positive Rate		10%		PPV 7,3%
True Positive Rate	(Se)	30%		
Prevalence		2,551		

Nu større prævalens - fordi vi kigger på både død og slagtet. Dermed bedre PPV

Scenarie C (3)		Sand tilstand		I alt
Udsat, Alarmtid: efter kælvning		Død	Leve	
Forudsagt tilstand	Død	15	95	110
	Leve	37	854	890
I alt		52	948	1.000
False Positive Rate		10%		PPV 13,6%
True Positive Rate	(Se)	29%		
Prevalence		5,163		

Scenarie D (4)		Sand tilstand		I alt
Udsat, Alarmtid: før kælvning		Død	Leve	
Forudsagt tilstand	Død	13	95	108
	Leve	39	853	892
I alt		53	947	1.000
False Positive Rate		10%		PPV 12,2%
True Positive Rate	(Se)	25%		
Prevalence		5,261		

Vores oprindelige kriterier for en 'god alarm' var en sensitivitet på mindst 50% og gerne på 60-70%. Og dertil en PPV på mindst 33% ved alarmer *efter* kælvning, og mindst 10% ved alarmer *før* kælvning.

Vi kan derfor give følgende bemærkninger til de fire scenarier:
Scenarie A opfylder ikke nogen af kriterier. Både sensitivitet og PPV er for lav.
Scenarie B har for lav sensitivitet, men PPV er 7,3%, så på den måde ligger den tæt på grænsen på de 10%, som vi krævede for alarmer *før* kælvning.
Scenarie C har også for lav sensitivitet, men nu med bedre PPV, nemlig på 13,6%, hvilket dog stadig er langt fra de 33%, som vi krævede for alarmer *efter* kælvning.

Scenarie D har igen en for lav sensitivitet, og faktisk dårligere end de andre, men til gengæld er PPV over de 10%, som vi krævede med alarmer *før* kælvning.

Dvs. at med scenarie D har vi en situation, hvor den ene parameter (sensitivitet) er for lav, men til gengæld er den anden parameter (PPV) højere end nødvendigt. De punkter for sensitivitet og specificitet, som er brugt som udgangspunkt for beregningen af ovenstående confusion matrices, er valgt hvor specificiteten er ca. 10%. Men vi kan også (frit) vælge andre steder på ROC-kurven (se ROC-kurven sidst i bilag 2).

Ved at afprøve forskellige punkter på ROC-kurven kan man udregne de tilsvarende confusion matrices og dermed også PPV. Vi kan aflæse forskellige punkter på ROC-kurven, bl.a. (24% , 48%) – (angivet ved mouse over i Azure ML?). En tilsvarende confusion matrix er:

Scenarie D (5)		Sand tilstand		I alt
Udsat, Alarmtid: før kælvning		Død	Leve	
Forudsagt tilstand	Død	25	227	253
	Leve	27	720	747
I alt		53	947	1.000
False Positive Rate		24%		PPV 10%
True Positive Rate	(Se)	48%		
Prevalence		5,261		

Vi sænker altså PPV til 10%, som er acceptabelt for en alarm før kælvning. Til gengæld får vi nu en sensitivitet på 48%. Og dermed er vi tæt på at opfylde målet om de mindst 50%.

Dermed har vi noget, som måske nærmer sig noget brugbart !...?...

Man skal her huske, at det datasæt, som er brugt som grundlag for disse beregninger, ikke havde alle de features med, som vi udpegede som ønskelige (og mulige). Det kan derfor ikke afvises, at ved at inddrage disse features kan vi få en bedre performance af alarmen. Disse features kunne fx være avlsværdier, besætningstype, goldperiodelængde og fedtsyresammensætning. Dermed kan der være håb om, at vi kan danne en alarm, som er tilstrækkelig god.

Der er dog en del ting, som herefter skal overvejes. Og disse forhold bliver beskrevet i næste afsnit.

Perspektiver, muligheder og begrænsninger

Mulige begrænsninger

Inden man springer ud i af ville bruge en ML-model ud fra ovenstående er der nogle punkter, som man måske først skal overveje:

- Vi har fået ovenstående tal fra output i Azure ML – og udledt nogle confusion matrices fra disse. Kan vi så også reelt få en model til at køre, som kan give disse resultater?
- Kan en model også performe lige så godt, hvis valideringsdatasættet bliver fx kælvninger i 2021? Det vil være et mere sandt valideringsdatasæt – og mere i

overensstemmelse med virkeligheden – end de automatisk udvalgte valideringsdatasæt, som dannes i Azure ML.

- De kriterier om Sensitivitet og PPV, som er beskrevet ovenfor, er kun et første bud. De skal måske revurderes og skal vel også afprøves i en reel situation, hvor landmænd kan vurdere alarmerne.
- Hvordan vil vi forholde os til tvillingefødsler? Disse fødsler ikke er med i ovenstående. Og det kan være en styrke for ovenstående resultater, fordi en tvillingefødsel er en ret god indikator for at koen er i risiko. Men den bedste alarm ovenfor er en alarm, som gives *før* kælvning. Og ofte ved man vel først vide, at der er tale om en tvillingefødsel, når fødslen er sket?
- Ovenstående arbejde er sket på data, som relaterer sig til 2. og senere kælvninger. Vi har ikke første kælvning med, og det er der nok heller ingen grund til, fordi risikoen for udsætning er væsentlige lavere for førstekalvs køer.
- Visse kritikere kunne måske hævde, at der ikke er grund til at lave avancerede ML-modeller. Det kunne måske påpeges, at man kunne lave en simpel og lige så god model ud fra tre af de vigtigste faktorer (features), fx Alder, goldydelse og kælvningsinterval. Det kunne måske undersøges og afkræftes.

Muligheder

Når man forholder sig til ovenstående mulige begrænsninger, skal man dog også forholde sig til at der er nogle forbedringsmuligheder ift. ovenstående.

- Kunne modellen ovenfor gøres bedre ved at inddrage nogle af de forklarende variable (features), som er nævnt ovenfor men ikke anvendt? Det er fx
  - avlsværdier (både koen og kalven)
  - besætning/besætningstype (hvilken slags? , øko?)
  - Goldperiodelængde
  - behandling i goldperioden
  - sygdomme i starten af sidste laktation
  - THI ved kælvning?
  - fedtsyresammensætning i slutning af laktation
  - men også vægt og huld ved goldning, selv om det nok er for færre køer, at vi har de oplysninger.

- Kunne modellen gøres bedre ved at udvide target, så vi får en højere prævalens - nemlig at inddrage visse sygdomme 0-7 dage efter kælvning? Evt. kunne man også sætte en grænse for død (og slagtet?) til 7 DEK? Dermed kan vi tale om "risiko for kælvningsproblemer"…? De sygdomme, som oftest ses efter kælvning er:
  - Børbetændelse,
  - Efterbyrd,
  - Yverbetændelse,
  - Ketose (først 7-10 DEK),
  - Kælvningsfeber

Ud fra resultaterne i de ovenstående modeller ser det du til, at vi får en bedre PPV ved at have en større prævalens – altså ved at vi vælger en risikohændelse, som ikke er alt for sjælden. Det ser vi i ovenstående, da vi går fra at betragte 'død' til at betragte både 'død og slagtet'. Det kunne pege i retningen af at fx inddrage sygdomme, som nævnt her. Og det leder også hen til, at det måske kunne være en fordel at se arbejdet her med 'Risiko for udsætning' i sammenhæng med den arbejdspakke, som har arbejdet med Risiko for ketose. Det *kan* være, at der kan opnås nogle fordele ved at betragte arbejdet samlet, men det kan også være, at der er for

stor forskel i de to tilgange, fx vægt-registreringer, som kun et fåtal af mælkeproducenterne har.

- Kan vi her også inddrage risiko hos kalven? – og altså ikke kun for koen…

- Man skal tage i betragtning, at Uheld/tilskadekomst vil vi næppe kunne forudsige, så det taler for at acceptere en let reduceret Sensitivitet. Ca. 20% af døde køer, som har en afgangsårsag, har Uheld/tilskadekomst som årsag.

Generelt kan man vælge (mindst) to forskellige tilgange til valg af features

a) Sørge for at de udvalgte features findes hos næsten alle individer/køer/kælvninger. Det betyder færre features, og måske dårligere prædiktion. Til gengæld kan alarmer beregnes for rigtig mange individer/køer/kælvninger
b) Arbejde med features, som er rigtig gode til at sige noget om problemstillingen, fx vægt og/eller huld ved goldning. Hvis disse features så viser sig at være værdifulde, så kan de motivere flere landmænd til at opsamle, registrere og få værdi af disse features. Til gengæld er det kun relativt få landmænd, som kan få beregnet en alarm.

Konklusion

Vi har i særlig grad kigget på risiko for uønsket udsætning (død, aflivning, slagtning) af en ko inden for 60 dage efter kælvning. Målet har været at kunne forudsige denne risiko. Dermed kan landmanden tage action allerede i forbindelse med, at koen skal kælve eller kort derefter.

Ved at bruge en række features (risikofaktorer), som allerede er indberettet til kvægdatabasen, har vi i Azure ML fundet en Machine Learning model, som næsten giver den præcision, som vi opstillede i løbet af gruppens arbejde. Præcision er her forstået som Sensitivitet og Positive Predictive Value (PPV). Vi har fået en PPV på 10% som ønsket, og en Sensitivitet på 48%.

Scenarie D (5)		Sand tilstand		I alt
Udsat, Alarmtid: før kælvning		Død	Leve	
Forudsagt tilstand	Død	25	227	253
	Leve	27	720	747
I alt		53	947	1.000
False Positive Rate		24%		PPV 10%
True Positive Rate	Sensitivitet	48%		
Prevalence		5,261		

Vi ønskede dog en sensitivitet på mindst 50% og gerne 60-70%, men der er stadig mulighed for bedringer, som angivet i afsnittet om ”Perspektiver, muligheder og begrænsninger”. I dette afsnit er også angivet nogle punkter, som man nok bør tjekke op på i et evt. videre arbejde.

Bilag 1: Resultat af fem runs - Screendumps fra Azure ML

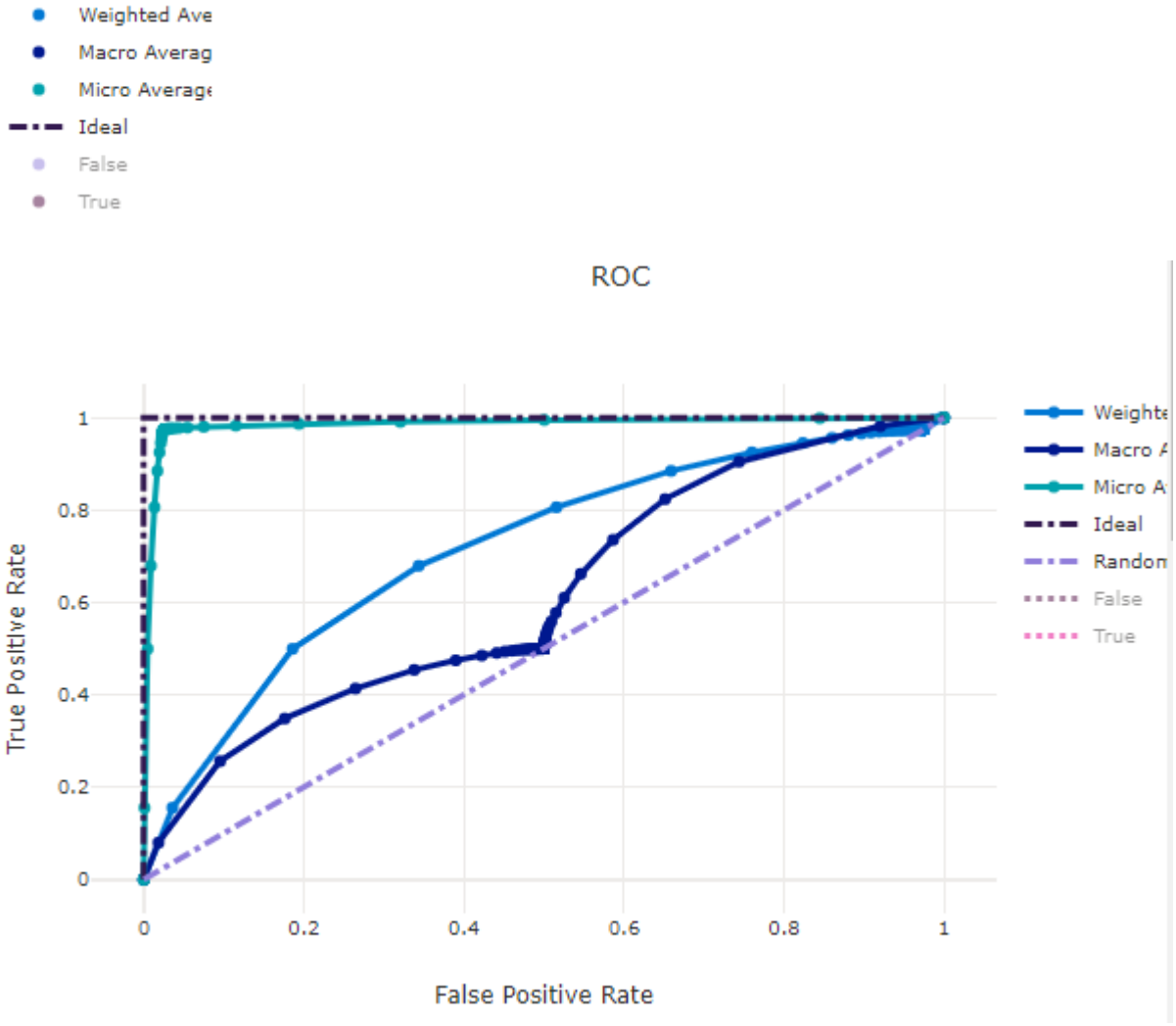
Alle modeller fokuserer på prædiktion af Død, med alarmtidspunkt *efter* kælvning.

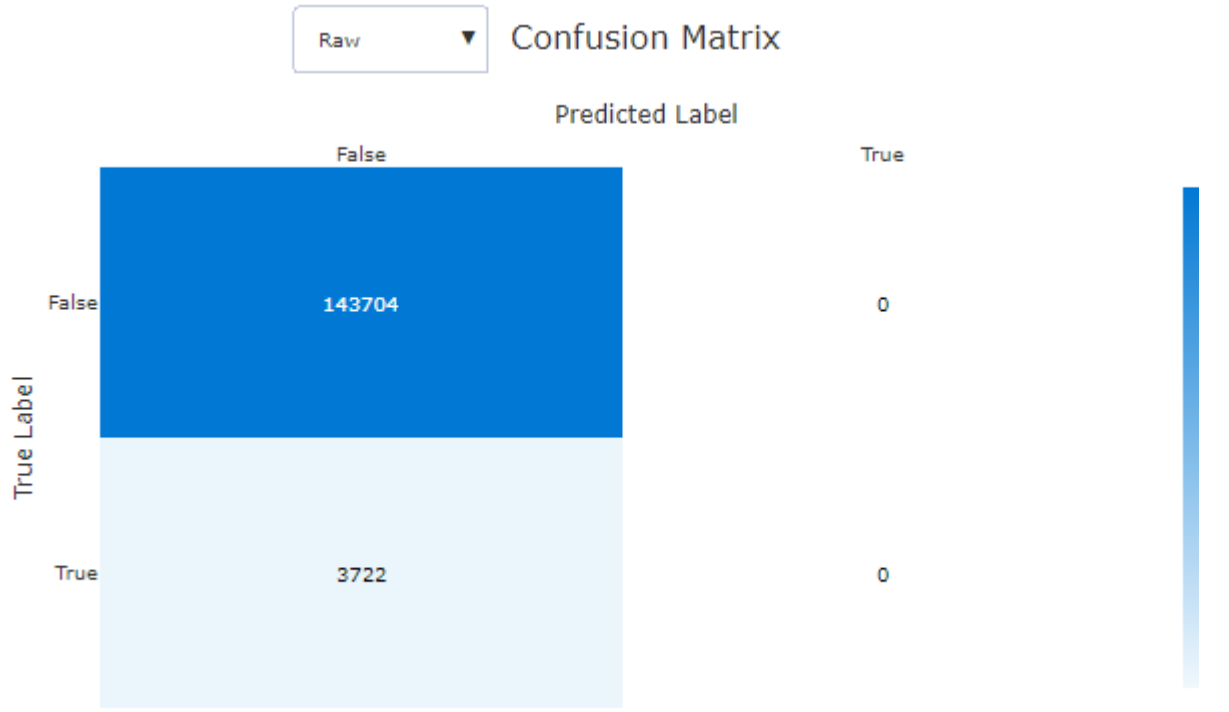
Nedenfor har jeg en række steder skrevet to procenter ved siden af hinanden, fx (10% , 33%). Det er et punkt fra ROC-kurven, hvor jeg har valgt et punkt med første tal tæt på 10%, altså med en specificitet på 90%. Og de 33% er dermed sensitiviteten, som kan opnås ved dette cut-off. Procenterne er blot aflæst visuelt fra graferne (mouse over funktion?), men giver en idé om, hvor vi er.

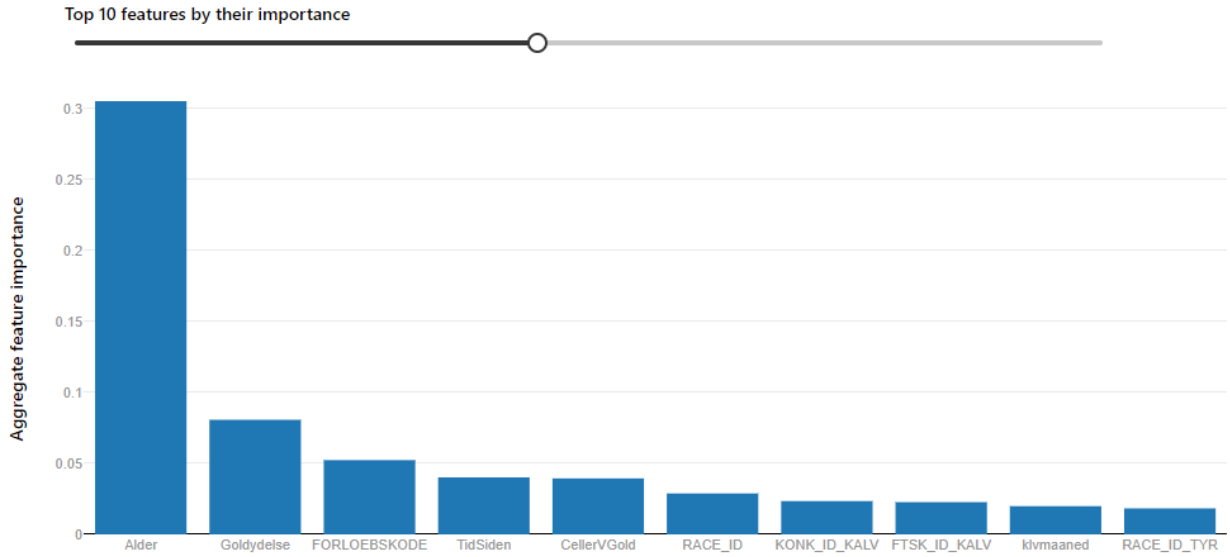
Bedste model fra Run 1

Algorithm name: VotingEnsemble

accuracy	AUC_binary	AUC_macro	AUC_micro	f1_score_macro	norm_macro_recall
0.975	0.731	0.731	0.986	0.494	0



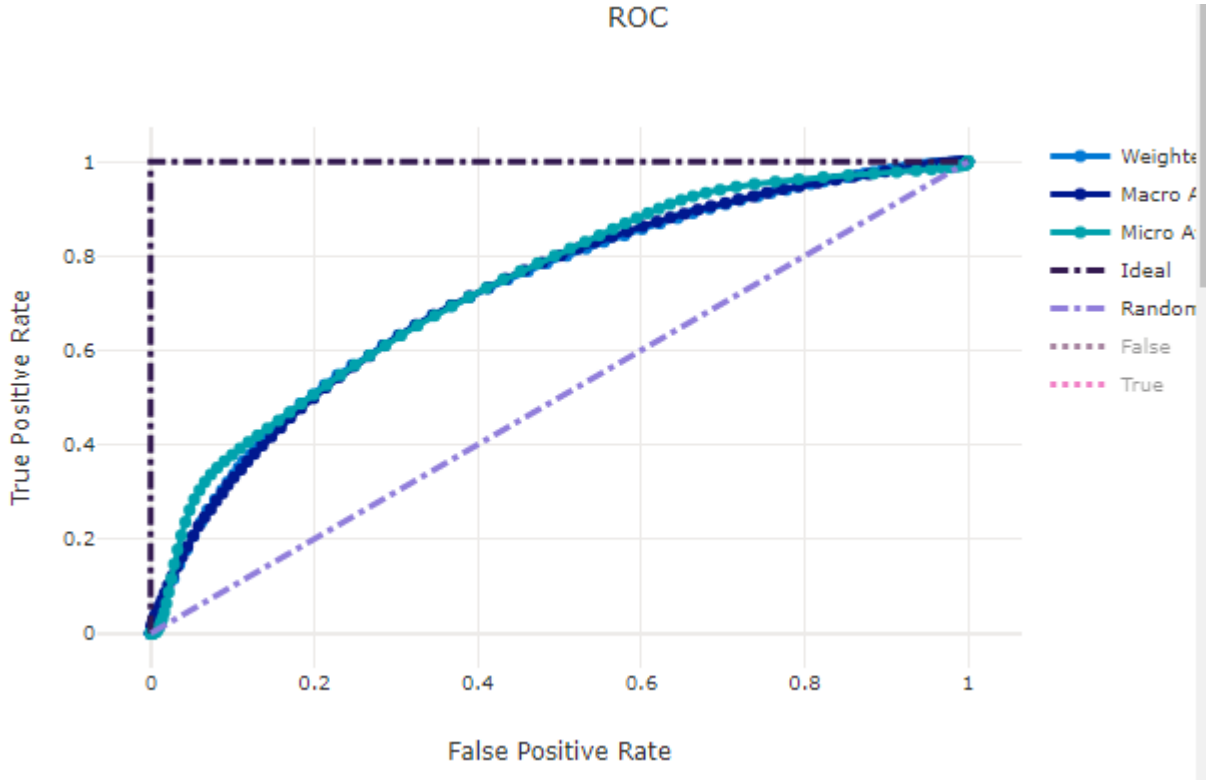




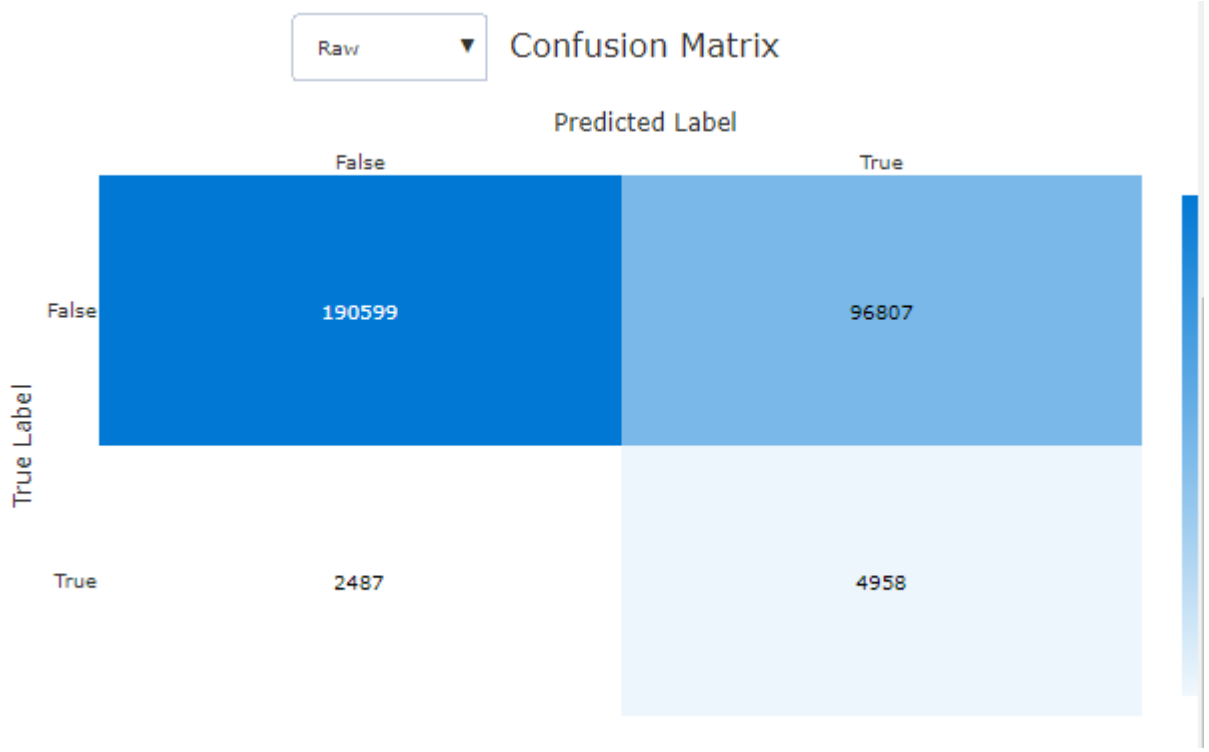
Bedste model fra Run 66

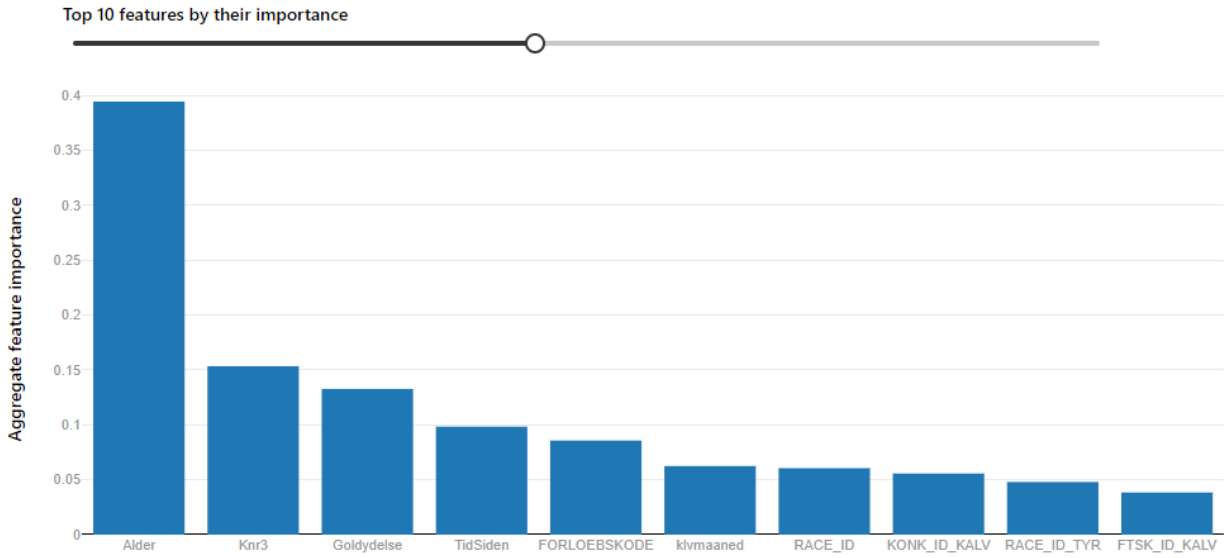
Algorithm name: MaxAbsScaler, SGD

accuracy	AUC_binary	AUC_macro	AUC_micro	f1_score_macro	norm_macro_recall
0.663	0.722	0.722	0.735	0.442	0.329


ROC
True Positive Rate
False Positive Rate
0
0.2
0.4
0.6
0.8
1
Weighte
Macro A
Micro A
Ideal
Randon
False
True

Fx (10% , 33%)

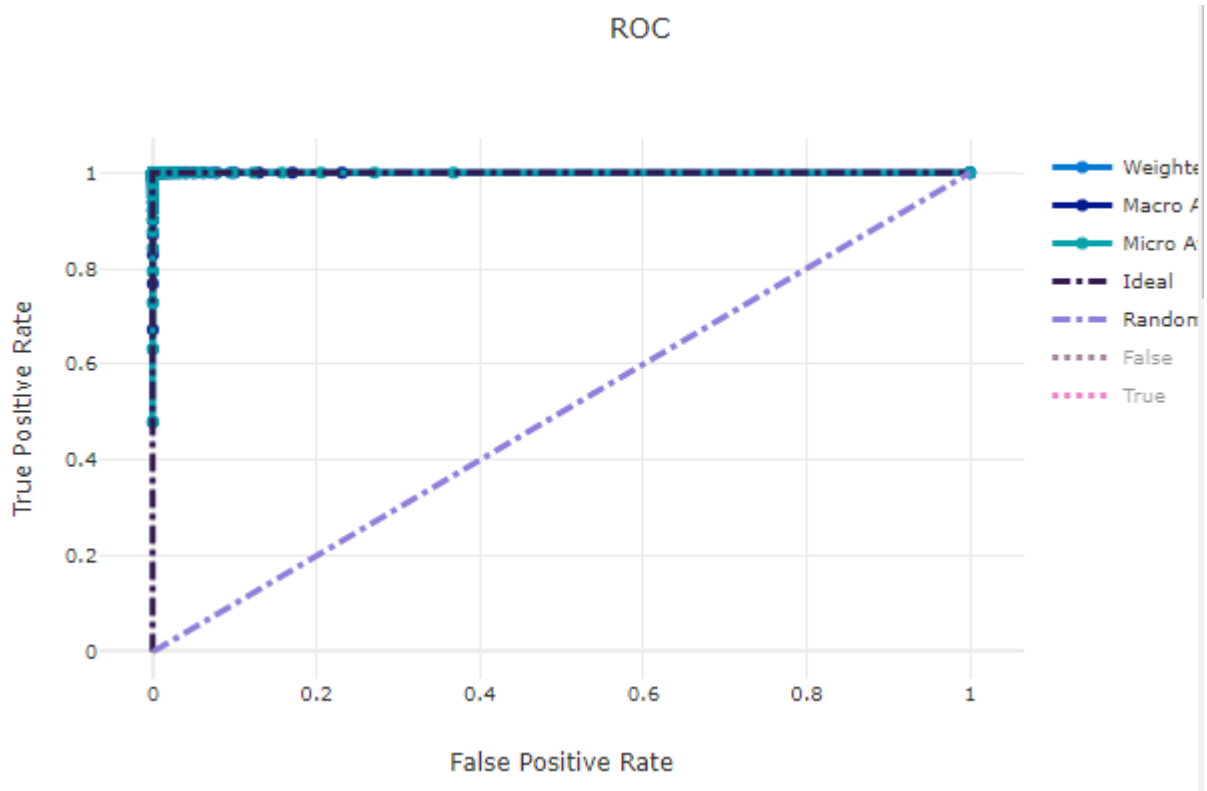

Raw
Confusion Matrix
Predicted Label
False
True
True Label
False
True
190599
96807
2487
4958



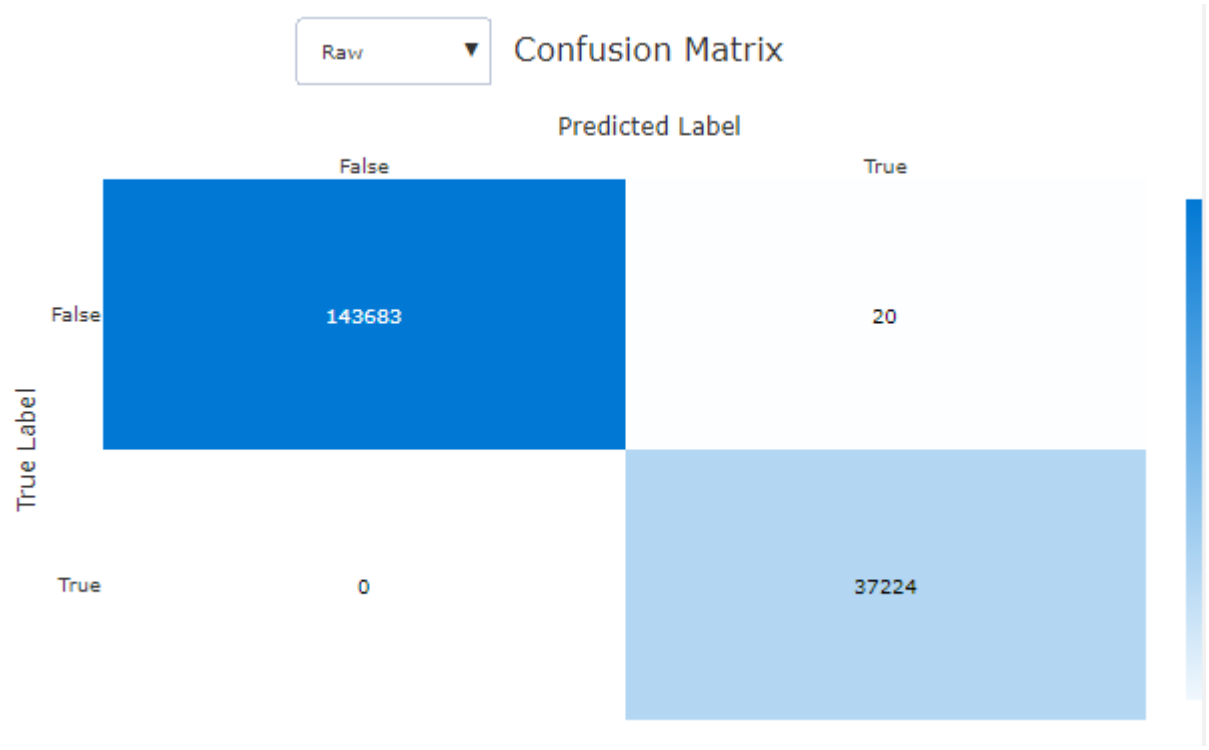
Bedste model fra Run 166

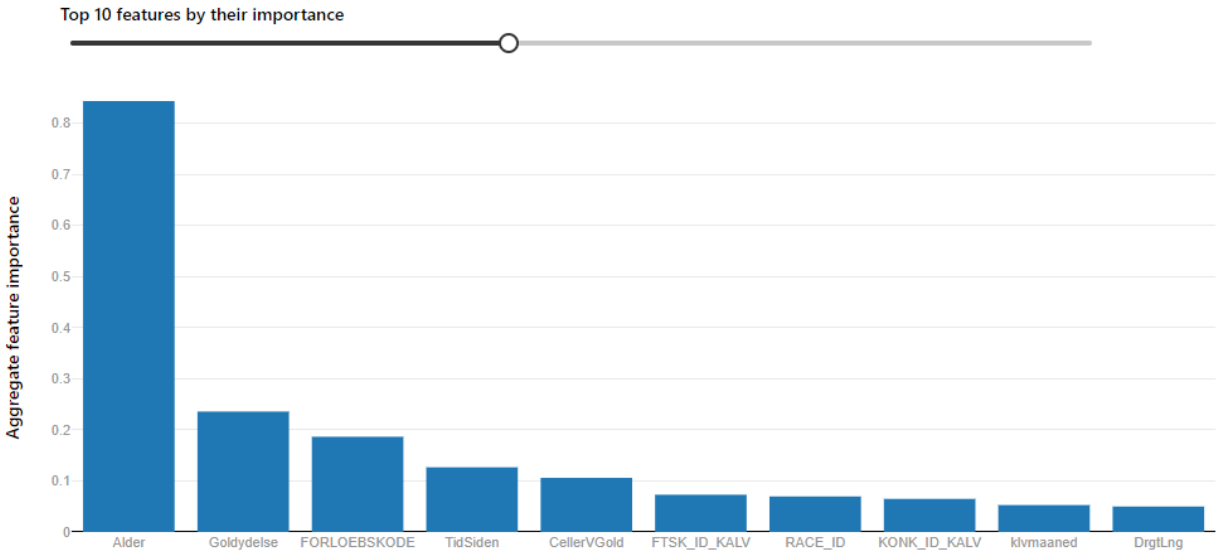
Algorithm name: StandardScalerWrapper, ExtremeRandomTrees

accuracy	AUC_binary	AUC_macro	AUC_micro	f1_score_macro	norm_macro_recall
1	1	1	1	1	1



Fx (0%, 100%)

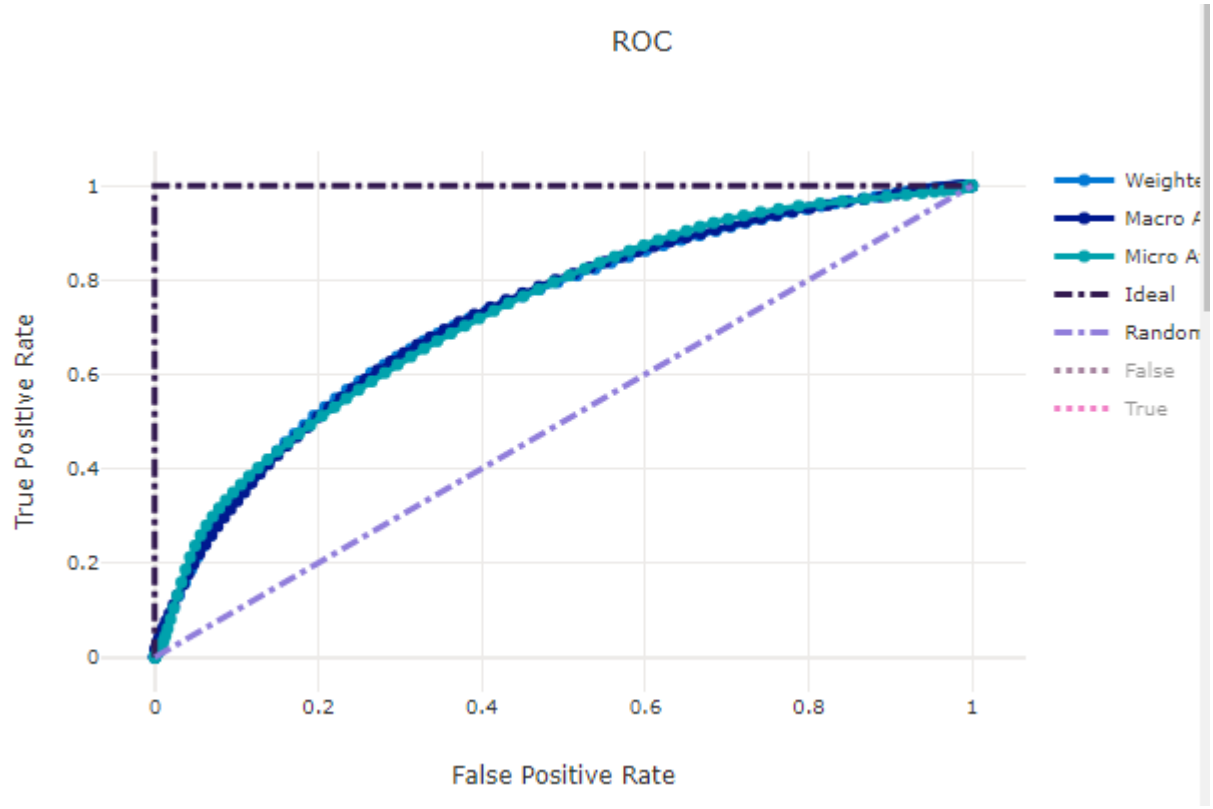




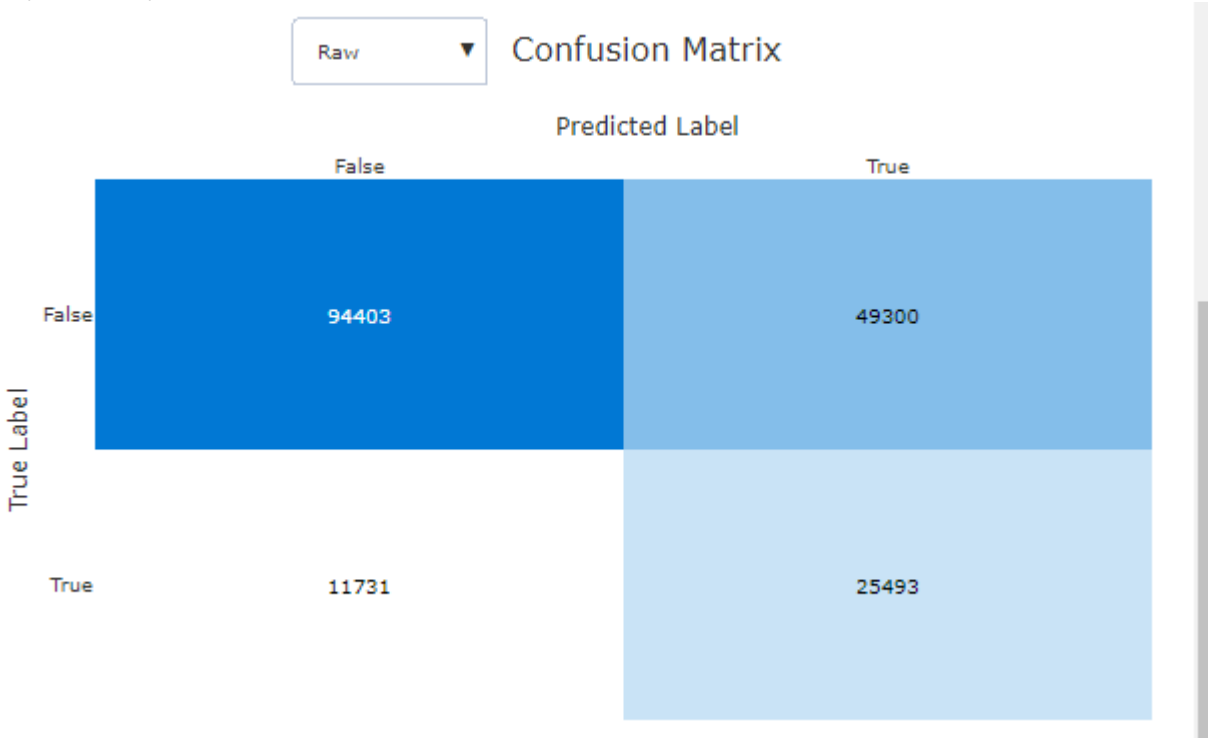
Næstbedste model fra Run 166

Algorithm name: StandardScalerWrapper, LogisticRegression

accuracy	AUC_binary	AUC_macro	AUC_micro	f1_score_macro	norm_macro_recall
0.663	0.727	0.727	0.73	0.605	0.342



Fx (10%, 33%)

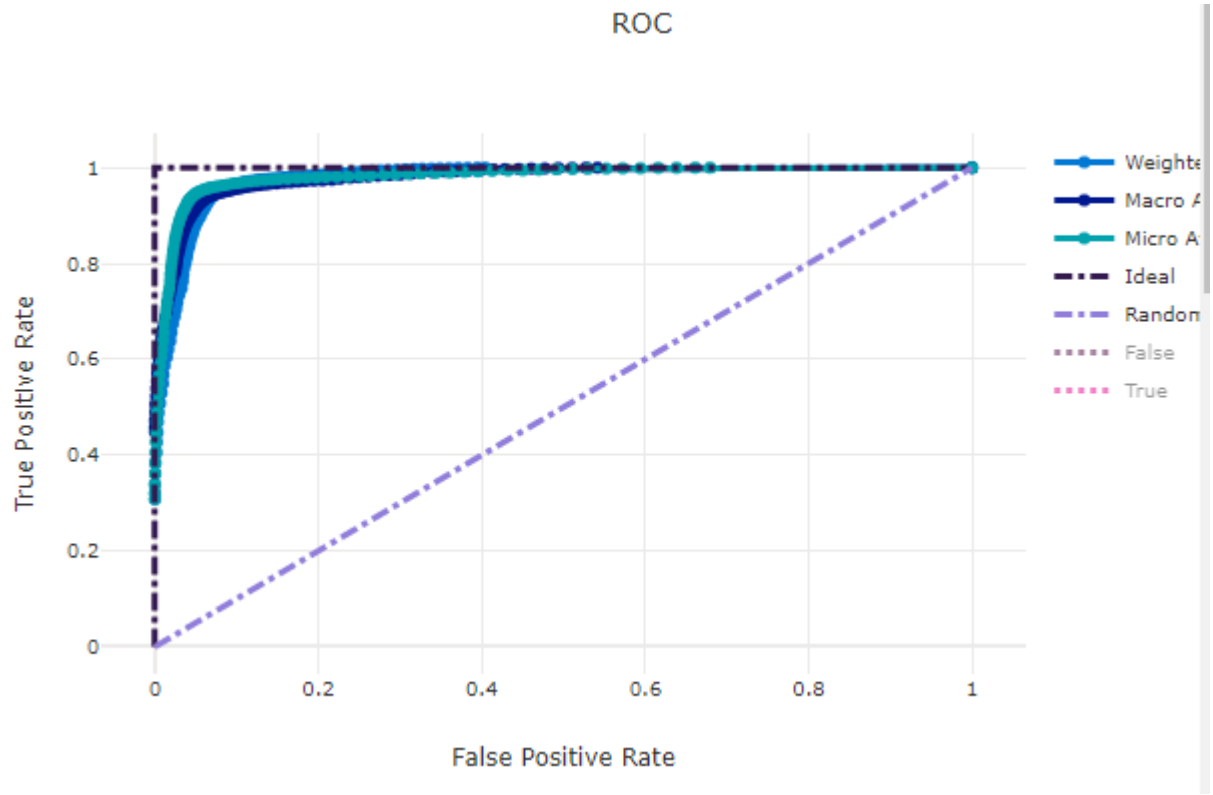


[Ingen Explanations]

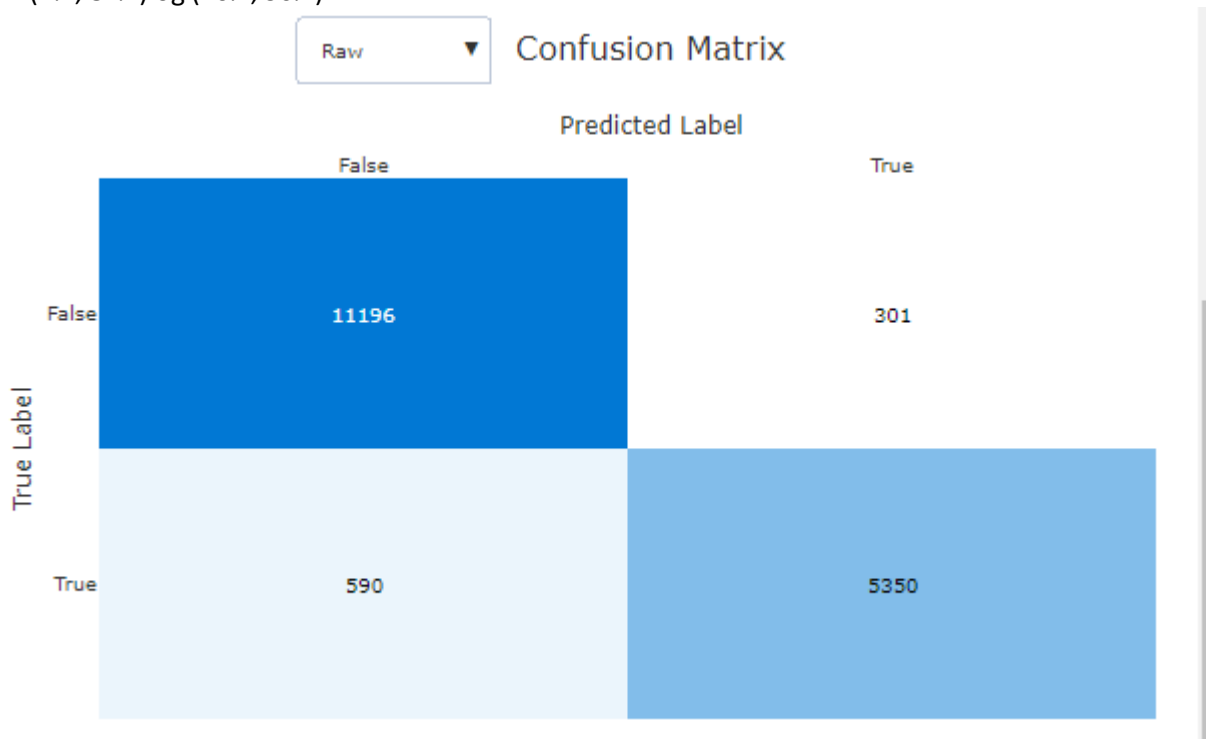
Bedste model fra Run 285

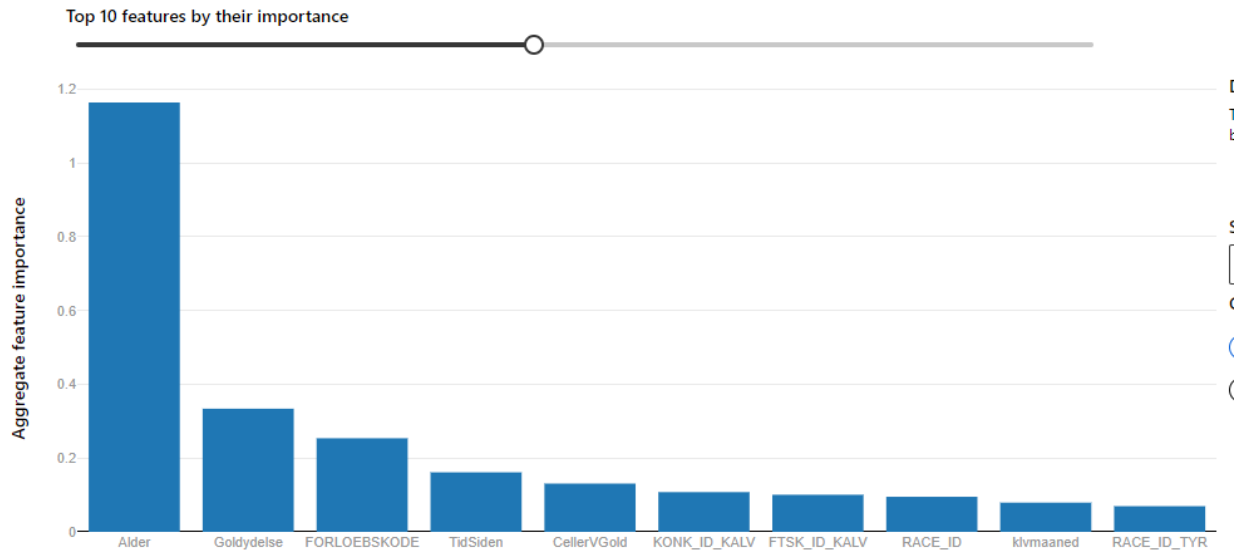
Algorithm name: StandardScalerWrapper, ExtremeRandomTrees

accuracy	AUC_macro	AUC_micro	f1_score_macro	norm_macro_recall
0.949	0.966	0.981	0.942	0.874



Fx (4%, 81%) og (10%, 96%)

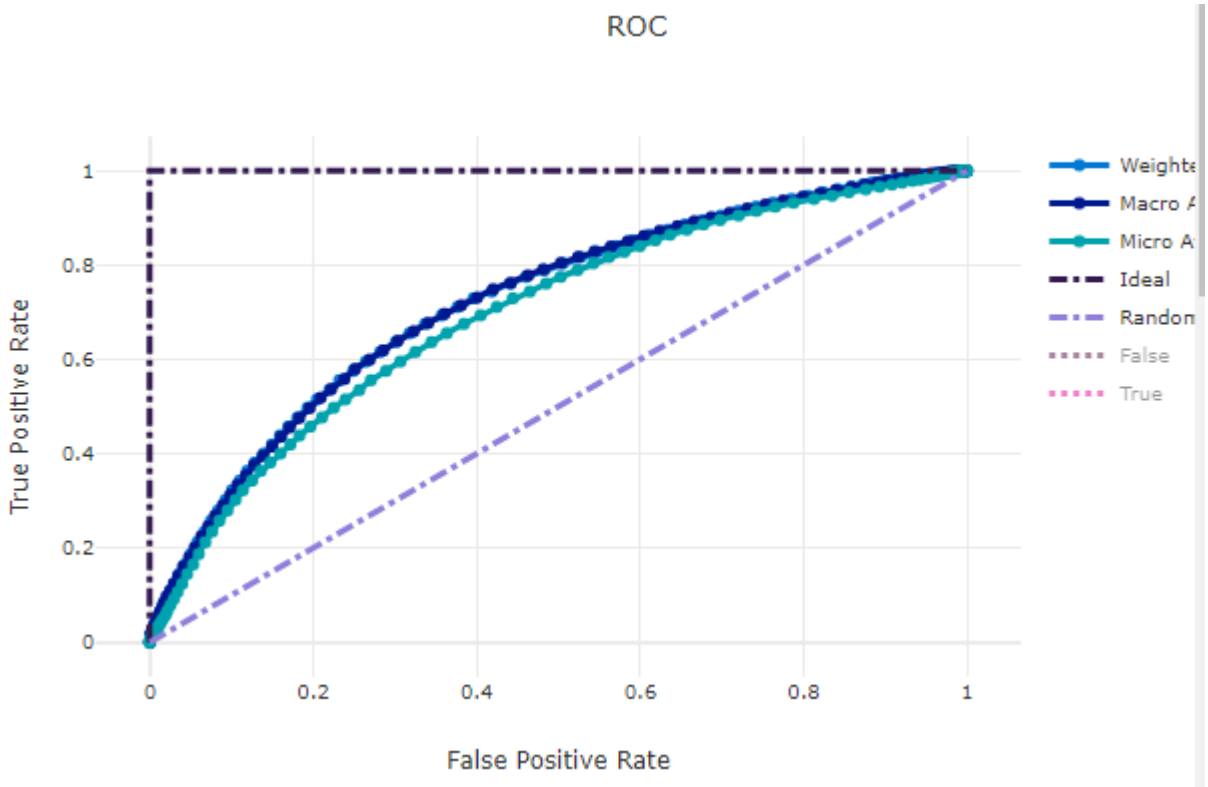




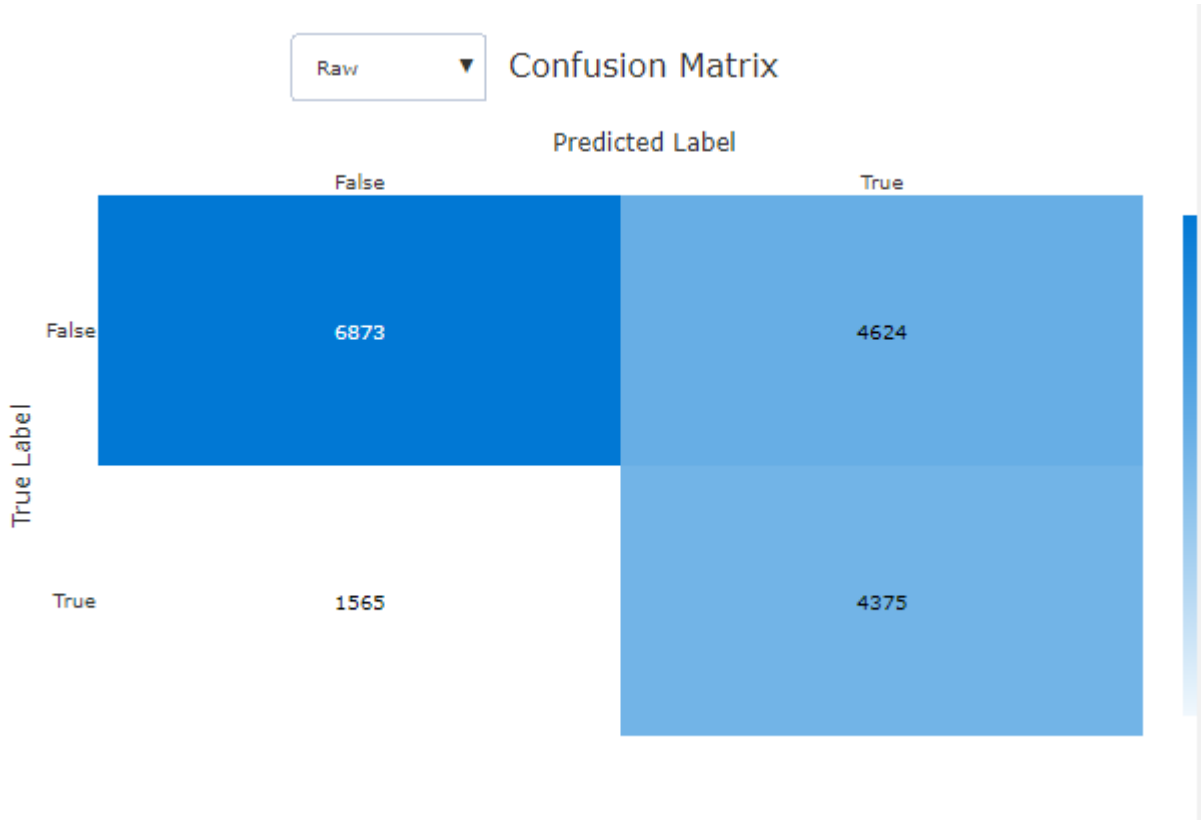
Næstbedste model fra Run 285

Algorithm name: MaxAbsScaler, SGD

accuracy	AUC_macro	AUC_micro	f1_score_macro	norm_macro_recall
0.645	0.724	0.697	0.638	0.334



Fx (10%, 30%)

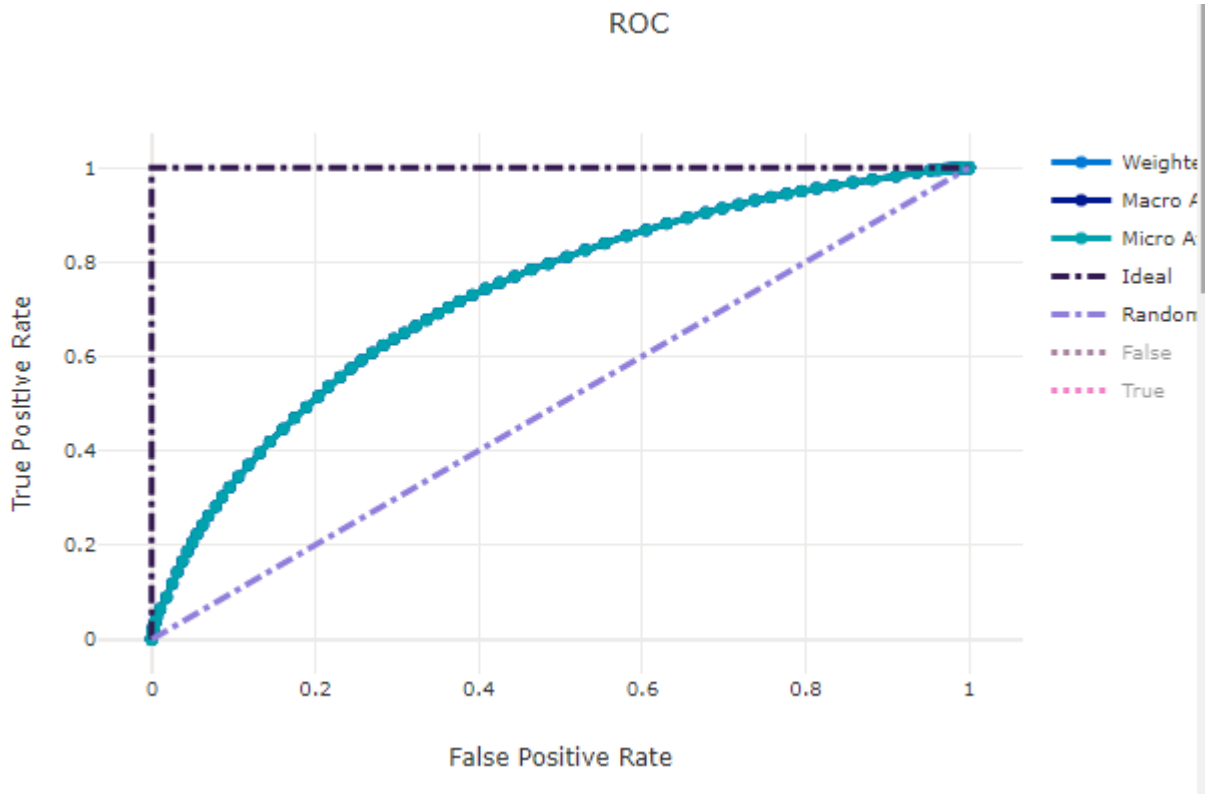


[Ingen Explanations]

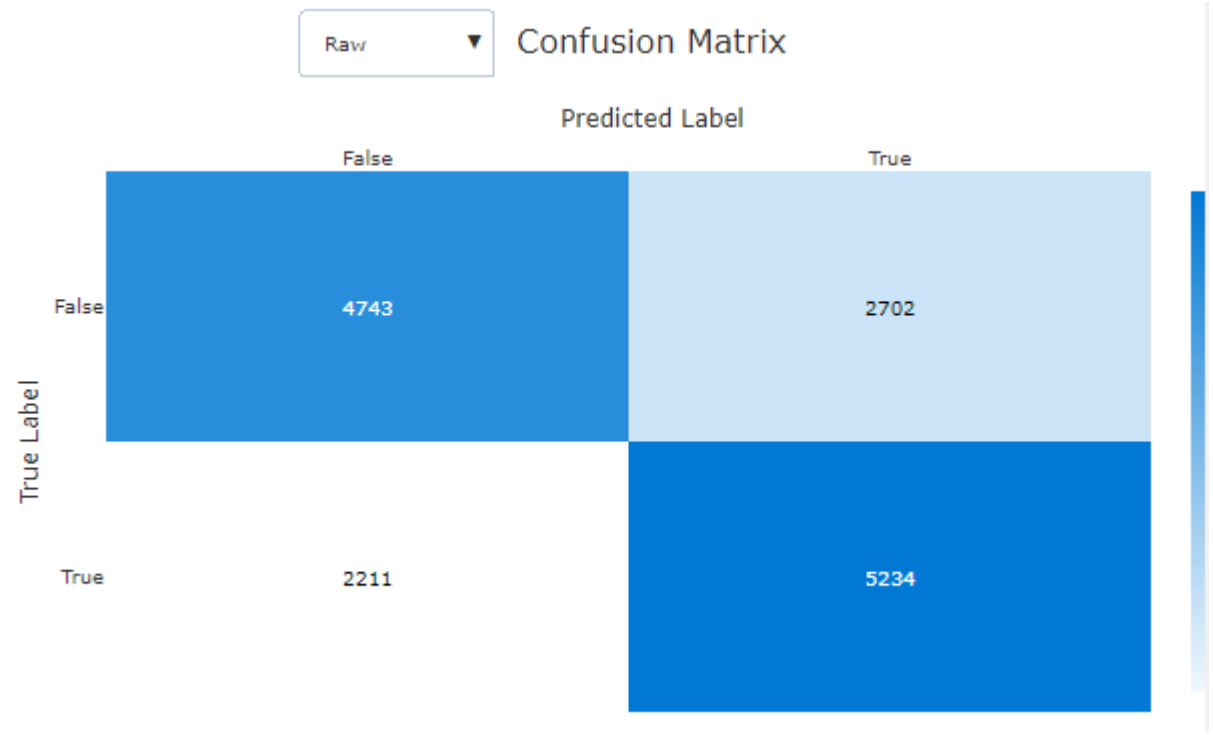
Bedste model fra Run 340

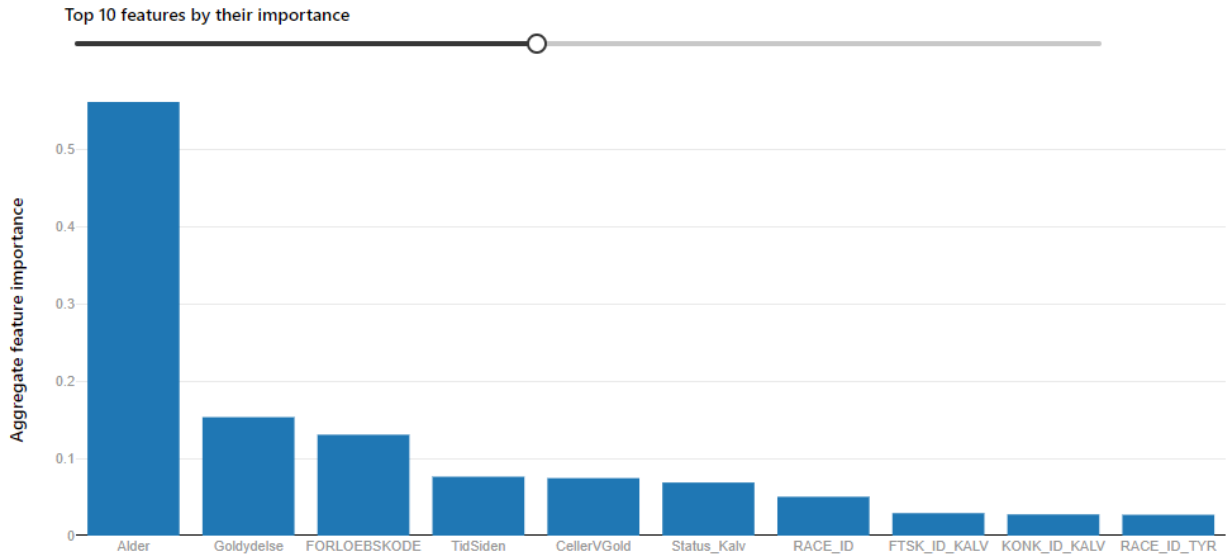
Algorithm name: MaxAbsScaler, XGBoostClassifier

accuracy	AUC_macro	AUC_micro	f1_score_macro	norm_macro_recall
0.67	0.728	0.728	0.67	0.34



Fx (10%, 34%)


Raw
Confusion Matrix
Predicted Label
False
True
False
True
True Label
False
4743
2702
True
2211
5234


Top 10 features by their importance
Aggregate feature importance
0.5
0.4
0.3
0.2
0.1
0
Alder
Goldydelse
FORLOEBSKODE
TidSiden
CellerVGold
Status_Kalv
RACE_ID
FTSK_ID_KALV
KONK_ID_KALV
RACE_ID_TYR

Bilag 2: Resultat af modeller fra fire scenarier - Screendumps fra Azure ML

Vi betragter de sidste tre af følgende fire modeller.

A. 2+ kælvning, 2015-2020, Død, Alarmtid: efter kælvning
B. 2+ kælvning, 2015-2020, Død, Alarmtid: før kælvning
C. 2+ kælvning, 2015-2020, Udsat, Alarmtid: efter kælvning
D. 2+ kælvning, 2015-2020, Udsat, Alarmtid: før kælvning

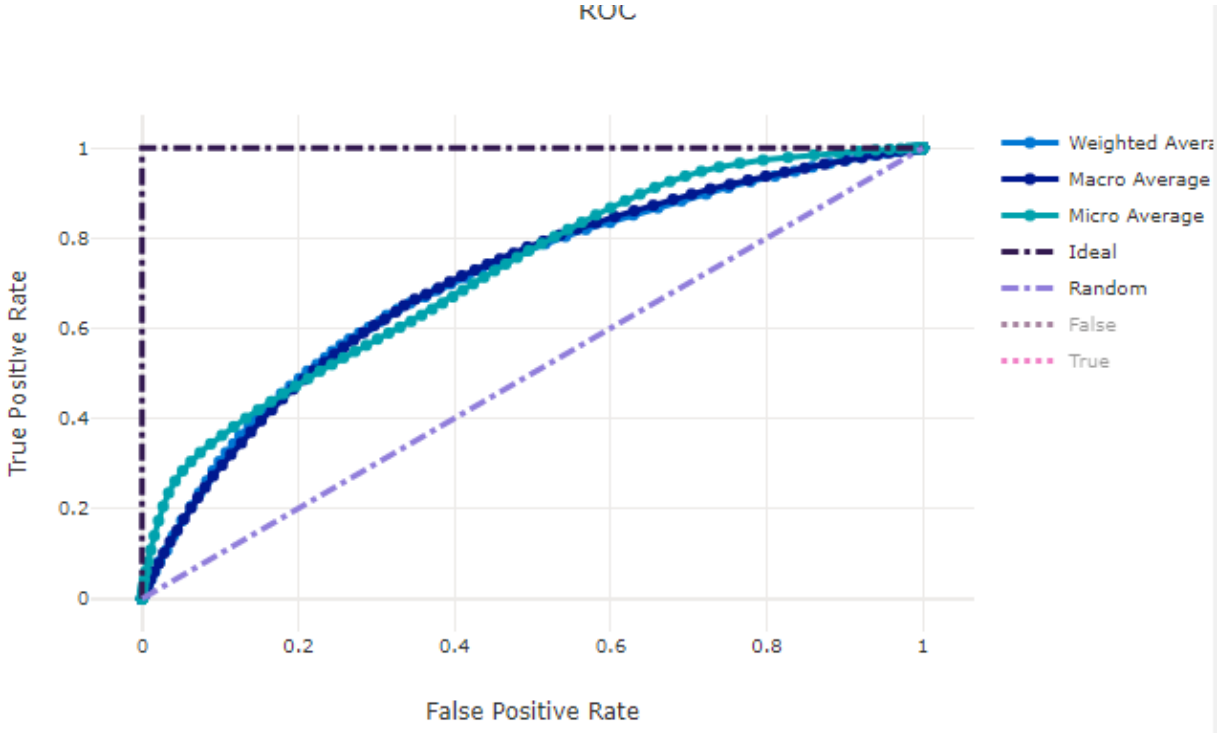
Scenarie A 2+ kælvning, 2015-2020, Død, Alarmtid: efter kælvning

Dette scenarie er uddybet i høj grad ovenfor.

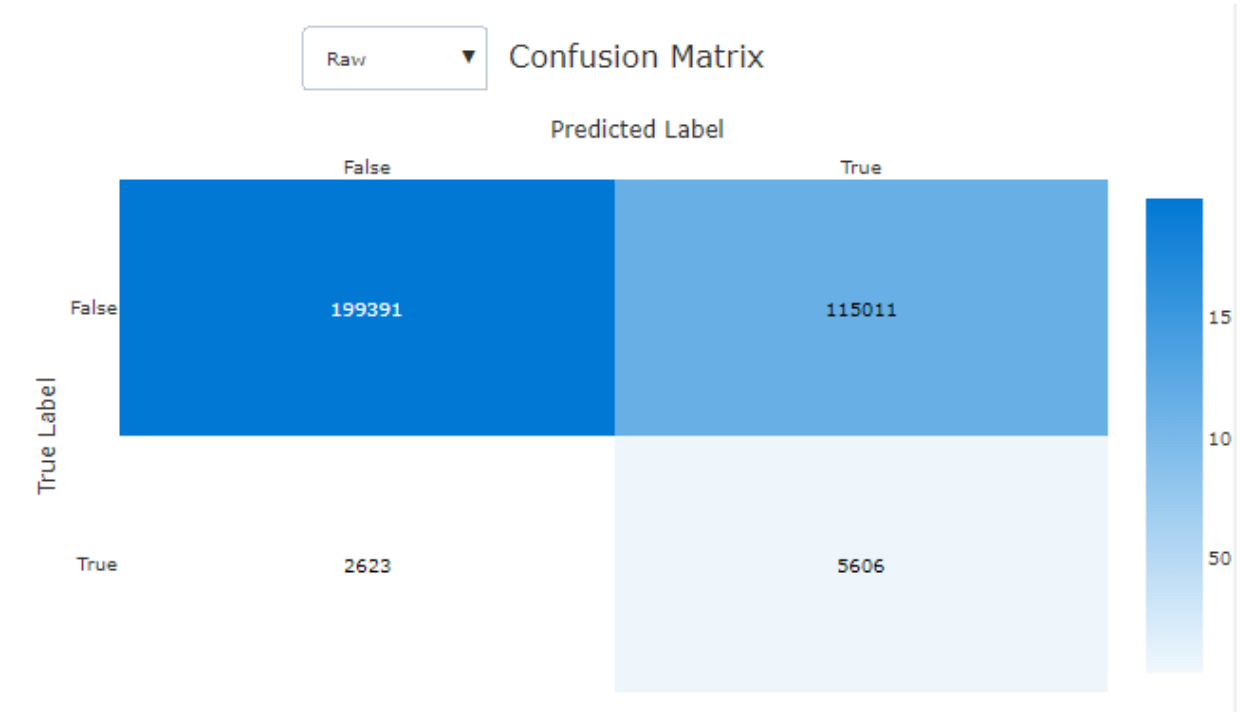
Scenarie B 2+ kælvning, 2015-2020, Død, Alarmtid: før kælvning

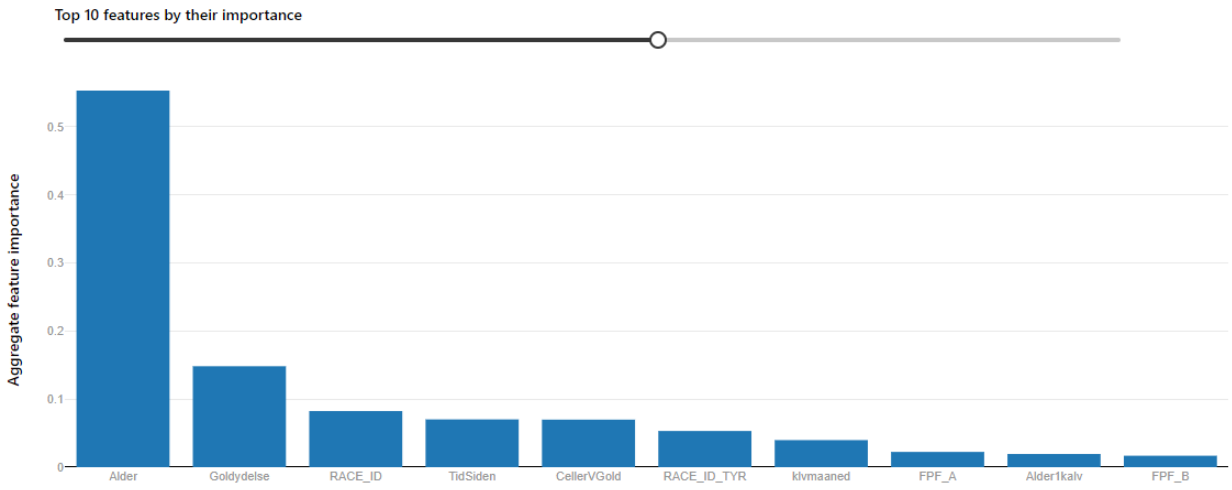
Algorithm name: MaxAbsScaler, LightGBM

accuracy	AUC_macro	AUC_micro	AUC_weighted	f1_score_macro	norm_macro_recall
0.635	0.704	0.721	0.704	0.43	0.315



Fx (10%, 30%)

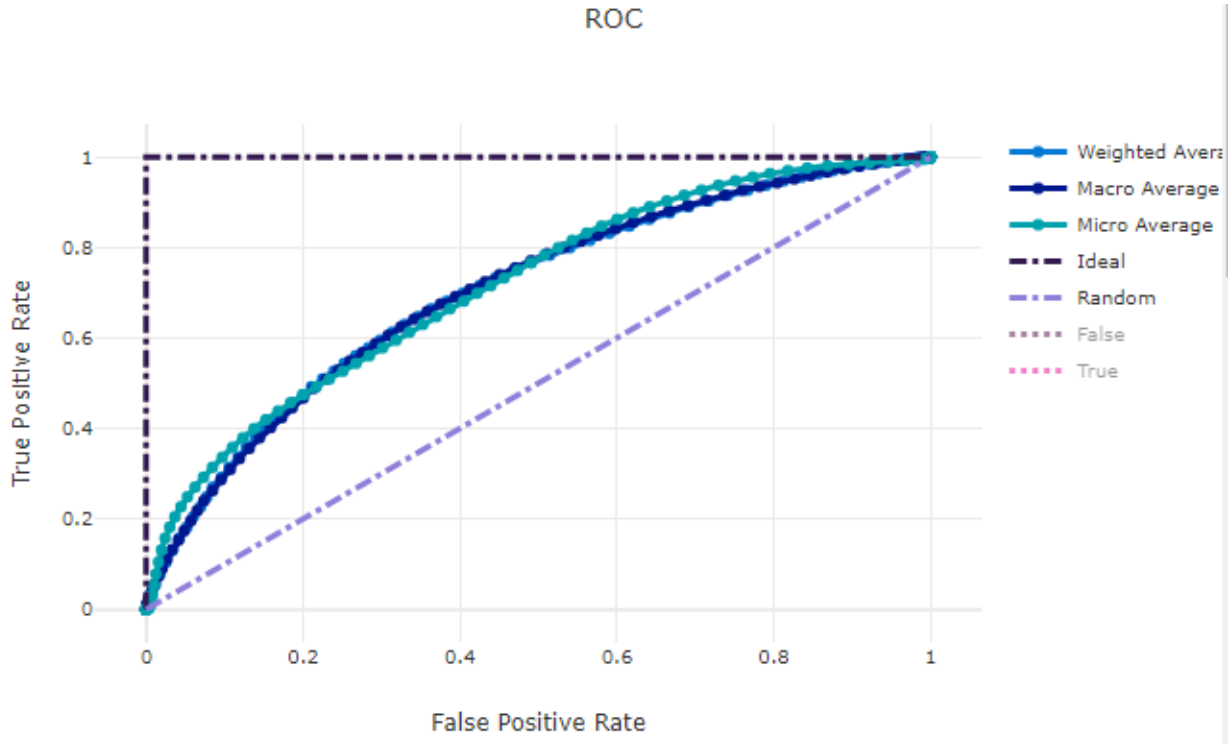




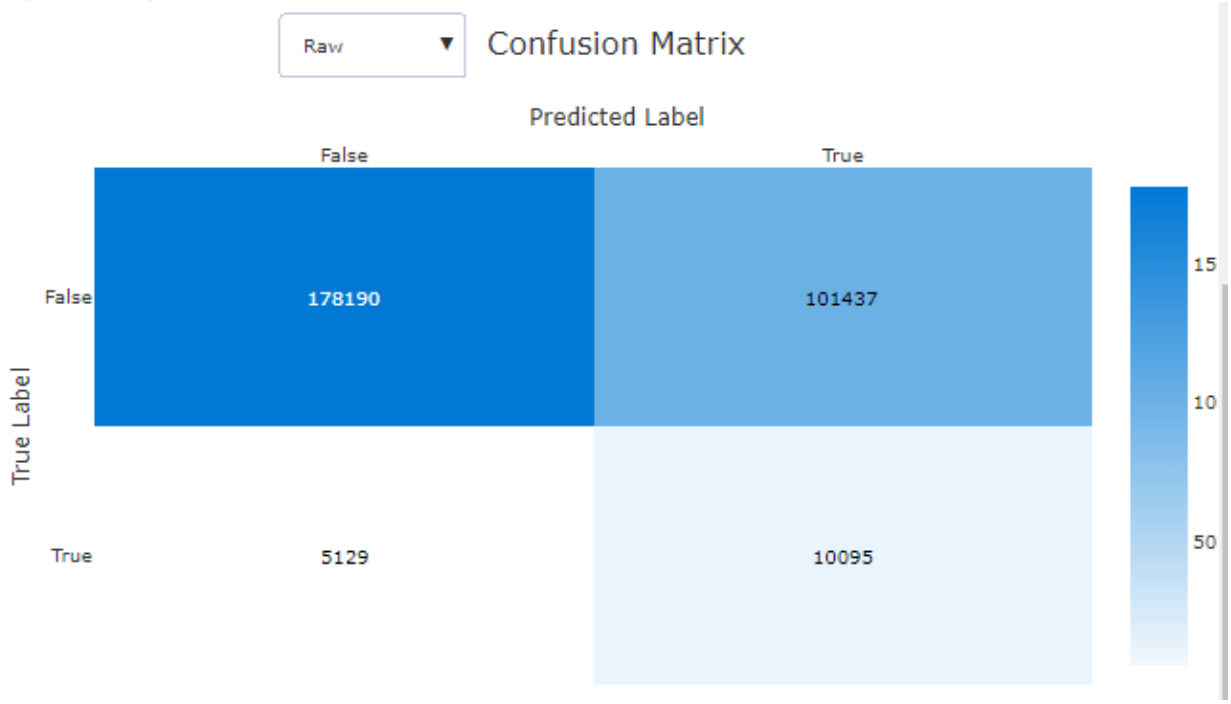
Scenarie C 2+ kælvning, 2015-2020, Udsat, Alarmtid: efter kælvning

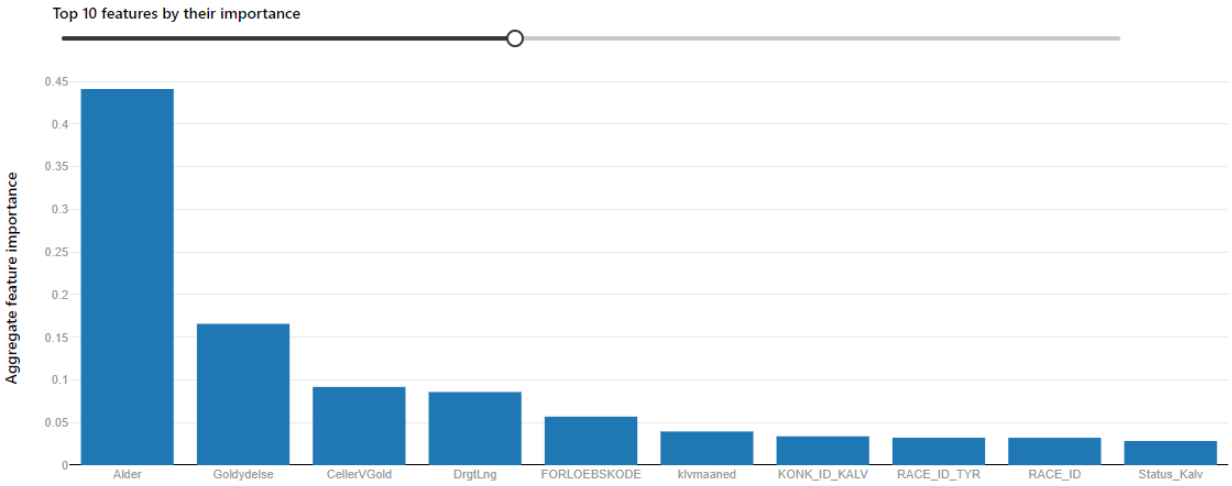
Algorithm name: MaxAbsScaler, LightGBM

accuracy	AUC_macro	AUC_micro	AUC_weighted	f1_score_macro	norm_macro_recall
0.639	0.703	0.714	0.703	0.465	0.3


ROC
True Positive Rate
False Positive Rate
0
0.2
0.4
0.6
0.8
1
Weighted Avera
Macro Average
Micro Average
Ideal
Random
False
True

Fx (10%, 29%)

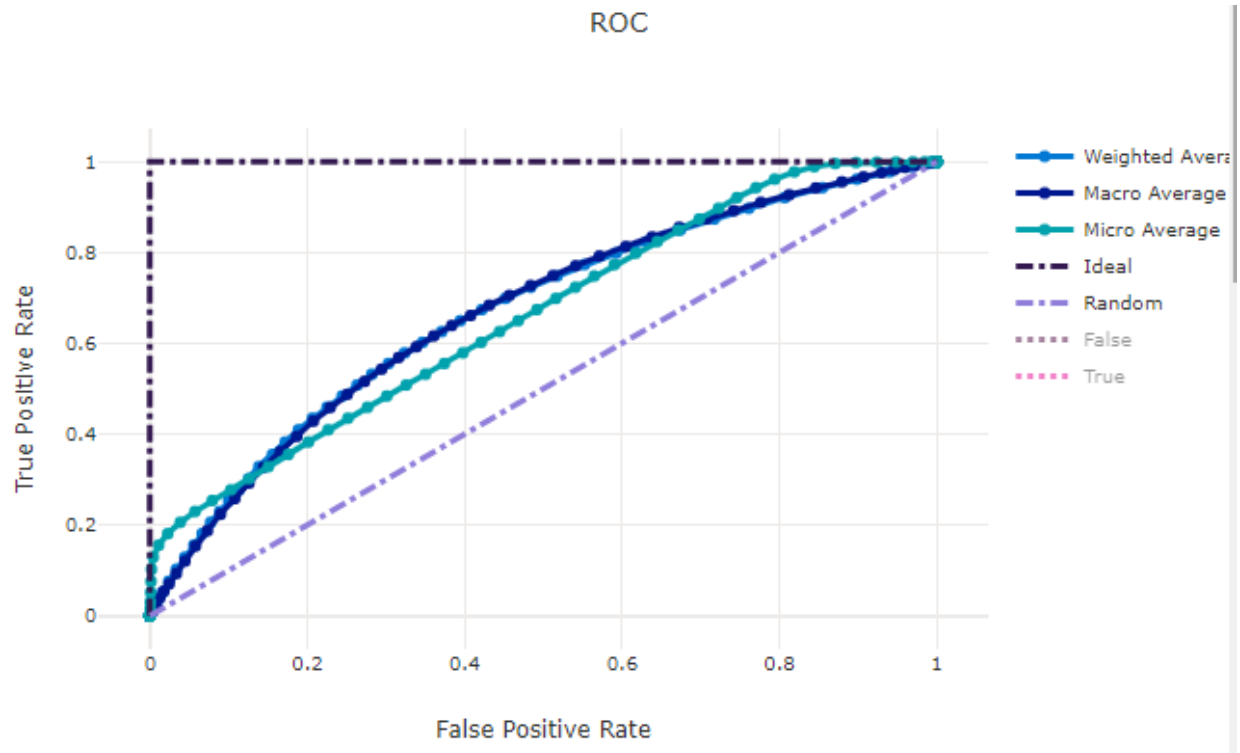

Raw
Confusion Matrix
Predicted Label
False
True
True Label
False
True
178190
101437
5129
10095
15
10
50



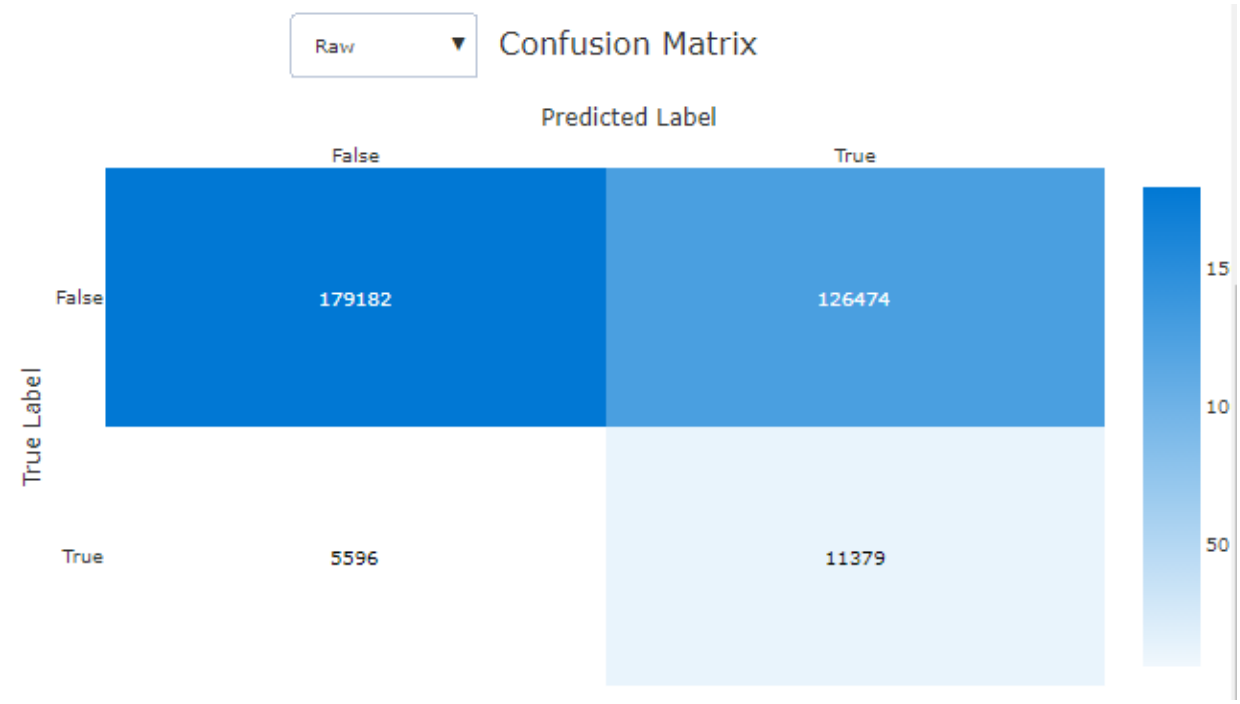
Scenarie D 2+ kælvning, 2015-2020, Udsat, Alarmtid: før kælvning

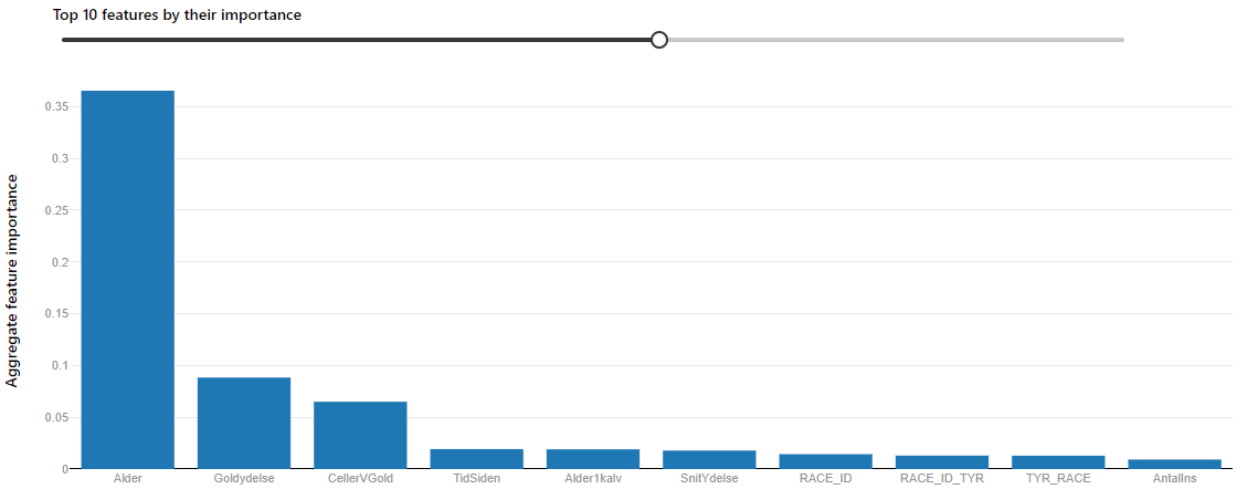
Algorithm name: TruncatedSVDWrapper, RandomForest

accuracy	AUC_macro	AUC_micro	AUC_weighted	f1_score_macro	norm_macro_recall
0.591	0.669	0.661	0.669	0.439	0.257



Fx (10%, 25%)


Raw
Confusion Matrix
Predicted Label
False
True
False
True
True Label
179182
126474
5596
11379
15
10
50


Top 10 features by their importance
Aggregate feature importance
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Alder
Goldydelse
CellerVGold
TidSiden
Alder1kalv
SnifYdelse
RACE_ID
RACE_ID_TYR
TYR_RACE
Antallns