VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

EKONOMETRIJOS BAKALAURO STUDIJŲ PROGRAMA

Rudųjų lapių krūminių dantų bei žandikaulio osteometrijos rezultatų analizė

Analysis of the red fox molar teeth and jaw osteometry results

Baigiamasis bakalauro darbas

Autorius: Neda Lapėnaitė
VU el. p.: neda.lapenaite@mif.stud.vu.lt

Darbo vadovas: Lekt., Dr. Rimantas Eidukevičius

Vilnius
2023

Santrauka

Šiame darbe yra daromos prognozės rudųjų lapių lyčiai ir žandikaulių ilgiui nustatyti pagal osteometrijos rezultatus. Modeliams sudaryti taikoma tiesinė regresija, logistinė regresija, sprendimų medžiai bei atsitiktinis miškas. Palyginus modelių rezultatus buvo nustatyti tiksliausi modeliai prognozuojant lytį ir žandikaulio ilgį. Duomenų analizė atlikta naudojantis R programavimo kalba, „Rstudio" sistemos Rattle paketu.

Raktiniai žodžiai: tiesinė regresija, logistinė regresija, sprendimų medis, atsitiktinis miškas, osteometrija, rudosios lapės.

Summary

In this work, predictions are made for the sex and jaw length of red foxes based on osteometry results. Linear regression, logistic regression, decision trees, and random forests are applied to build models. By comparing the results of the models, the most accurate models for predicting sex and jaw length were identified. Data analysis was performed using the R programming language and the Rattle package in the RStudio software.

Keywords: linear regression, logistic regression, decision tree, random forest, osteometry, red fox.

Turinys

1 Įvadas

Rudosios lapės yra dažnai sutinkamas gyvūnas Lietuvos teritorijoje. Šie gyvūnai garsėja savo sumanumu ir puikiomis prisitaikymo prie aplinkos žiniomis. Osteometrijos dėka mes galime gauti naudingos informacijos ne tik apie žmogaus kūną, bet ir gyvūnus. Kaulų matavimai mums suteikia vertingos informacijos, kuri padeda populiacijos, ekologiniams ar evoliuciniams tyrimams. Kadangi pasitaiko situacijų, kai turimi duomenys apie lapės dantis yra tik jų matavimai, šio darbo tikslas yra prognozuoti lapės lytį ir žandikaulio ilgį pasikliaujant turima informacija. Analizei atlikti buvo naudota R programavimo kalba naudojantis „Rstudio" sistema.

Darbas suskirstytas į dvi dalis: teorinę ir tiriamąją. Teorinėje dalyje aprašomas tiriamasis objektas, apžvelgiama literatūra ir taip pat trumpai paaiškinami naudojami terminai. Tiriamojoje dalyje aprašomi turimi duomenys, apžvelgiama aprašomoji statistika ir sudarinėjami modeliai išsikeltam tikslui pasiekti.

Darbo uždaviniai:

- Pritaikant tiesinę regresiją prognozuoti žandikaulio ilgį pagal viršutinio ir apatinio žandikaulių krūminių dantų osteometrijos rezultatus.
- Sudaryti sprendimų medį prognozuoti žandikaulio ilgį pagal viršutinio ir apatinio žandikaulių krūminių dantų osteometrijos rezultatus.
- Pritaikant logistinę regresiją prognozuoti lapės lytį pagal viršutinio ir apatinio žandikaulių krūminių dantų osteometrijos rezultatus.
- Sudaryti sprendimų medį prognozuoti lapės lytį pagal viršutinio ir apatinio žandikaulių krūminių dantų osteometrijos rezultatus.
- Sudaryti atsitiktinį mišką prognozuoti lapės lytį pagal viršutinio ir apatinio žandikaulių krūminių dantų osteometrijos rezultatus.

2 Teorinė dalis

2.1 Temos aprašymas

2.1.1 Rudosios lapės

Rudosios lapės (lotyniškai vadinamos Vulpes vulpes) yra ypač puikiai prie aplinkos prisitaikantys žinduoliai, gyvenantys įvairiose buveinėse visame pasaulyje, įskaitant miškus, pievas, kalnus, dykumas ir net žmonių aplinką. Vidutiniškai šie žinduoliai gyvena nuo dviejų iki keturių metų. Lapės ilgis nuo 46 centimetrų iki 86 centimetrų, o jų uodegos siekia nuo 30 centimetrų iki 55 centimetrų. Jų svoris kinta nuo 3 kilogramų iki 11 kilogramų.

Šios lapės žinomos dėl savo išradingumo ir yra įgudusios medžiotojos, mintančios graužikais, vištomis, triušiais, paukščiais ir kitais smulkiais gyvūnais. Jų mityba gali būti lanksti, įskaitant vaisius, daržoves, žuvį, varles, kirminus ir net žmonių aplinkoje rastą maistą, kaip atliekos ar naminių gyvūnų maistas. Rudosios lapės turi storą uodegą, kuri padeda išlaikyti pusiausvyrą ir šildo šaltu oru.

Lapės turi sugebėjimą dalintis informacija kvapu, kai jos pažymi savo buvimą šlapindamosi ant medžių ar uolų. Šių gyvūnų poravimosi sezonas yra žiema. Patelė atsiveda nuo 2 iki 12 jauniklių. Iš pradžių jaunikliai gimsta su rudu arba pilku kailiu, o spalva išryškėja per pirmąjį mėnesį. Abu tėvai prižiūri jauniklius, kol rudenį jie tampa savarankiški.

Nors rudosios lapės retkarčiais medžioja tik dėl sporto, tai nėra plačiai paplitusi praktika. Jos taip pat gali būti laikomos kenkėjais arba pasiutligės nešiotojomis tam tikrose situacijose. Apskritai, rudosios lapės demonstruoja savo prisitaikymą, intelektą ir gudrumą, leidžianti joms klestėti įvairiose aplinkose ir įtvirtinti savo buvimą visame pasaulyje [1].

2.1.2 Literatūros apžvalga

2022 metais buvo atliktas tyrimas pavadinimu „Rudųjų lapių krūminių dantų morfometrinė analizė Lietuvoje" (angl. „Morphometric analysis of the red fox molar teeth in Lithuania" [2]). Tyrimo autoriai Eugenijus Jurgelėnas, Indrė Jasinevičiūtė, Sigita Kerzienė ir Linas Daugnora. Jų tyrimo tikslas buvo ištirti rudųjų lapių viršutinio ir apatinio žandikaulio krūminių dantų ilgius, pločius bei žandikaulių (dantų eilių) ilgius ir išsiaiškinti ar pagal šiuos matmenis galima identifikuoti ar tiriamasis objektas yra patinas ar patelė. Duomenims gauti buvo atlikta osteometrinė analizė A. von den Driesch (1976) metodu. Matavimai buvo atlikti 230 kaukolių. 113 patinų ir 117 patelių.

Tyrimo metu buvo išsiaiškinta, kad daugumos patinų osteometrijos rezultatai buvo didesni nei patelių. Buvo stebėta koreliacija tarp dantų matavimų ir žandikaulio ilgio. Įvertinus koreliaciją tarp matavimų paaiškėjo, kad rezultatai kinta nuo labai silpno iki vidutinio stiprumo koreliacijos. Wilks' lambda analizė parodė rezultatus intervale nuo 0,821 iki 1, todėl buvo padaryta išvadą, kad dantų matavimo tyrimas yra nepatikimas rudųjų lapių lyčiai nustatyti [2].

Šio tyrimo tikslas duomenims pridėti papildomų kintamųjų ir padaryti prognozę ne tik lyties nustatymui, bet ir žandikaulio ilgiui. Tikimasi, jog papildomi kintamieji bus naudingi šioms prognozėms atlikti.

2.2 Terminai

ROC kreivė

ROC kreivė (angl. Receiver operating characteristic) yra grafikas, kuris rodo klasifikuoto modelio veikimą visuose klasifikavimo slenksčiuose. Norint apskaičiuoti taškus ROC kreivėje, įvertinamas logistinės regresijos modelis su skirtingais klasifikavimo slenksčiais. Praktikoje patogu naudotii plotą po kreive, kuris vadinamas AUC [3].

AUC

AUC (angl. Area under the curve) - Plotas po kreive. AUC parodo modelio efektyvumą. Kai AUC mažesnis nei 0.70, modelis yra nepatikimas. Nuo 0.70 iki 0.80 - modelis veikia pakankamai gerai. Kai AUC didesnis nei 0.80, modelis veikia puikia, o kai AUC reikšmė yra 1.00, tai reiškia tobulą modelio veikimą [4].

Sprendimų medis

Sprendimų medis (angl. Descition tree) yra vienas iš galingiausių mašininio mokymo algoritmų įrankių, naudojamų tiek klasifikavimo, tiek regresijos uždaviniams spręsti. Jis sukuria medžio tipo struktūrą, kur kiekviena viršūnė nurodo kintamojo testą, kiekviena šaka atspindi testo rezultatą, o lapas rodo klasę. Medis konstruojamas rekursyviai skaidant mokymo duomenis į poaibius, remiantis kintamųjų reikšmėmis, kol pasiekiama sustojimo sąlyga, pvz., medžio maksimalus gylis arba minimalus mėginio skaičius, reikalingas norint suskaidyti viršūnę.

Mokymo metu sprendimų medžio algoritmas atrenka geriausią kintamąjį, pagal kurį suskaidyti duomenis, remiantis metrika, tokia kaip entropija, kuri matuoja poaibių atsitiktinumą, arba Gini priemaišos (angl. Gini impurity), kurios matuoja poaibiuose esančias netinkamas reikšmes. Tikslas yra rasti kintamąjį, kuris maksimizuoja informacijos prieaugį arba netinkamų reikšmių sumažėjimą po skaidymo [5].

Tiesinė regresija

Tiesinė regresija yra paprastas ir dažnai naudojamas statistinis metodas, naudojamas mašininio mokymosi nuspėjamai analizei. Šiame analizės būde daroma prielaida, kad tarp nepriklausomo kintamojo ir priklausomo kintamojo yra tiesinis ryšys ir siekiama rasti geriausiai tinkančią liniją, apibūdinančią šį ryšį. Tiesinės regresijos lygtis:

$$Y_i = \beta_0 + \beta_1 X_i,$$

kur Y_i yra priklausomas kintamasis arba numatoma reikšmė, β_0 laisvasis narys, dar vadinamas konstanta (angl. intercept), β_1 krypties koeficientas (angl. slope) ir X_i yra nepriklausomas kintamasis [6].

Determinacijos koeficientas

Determinacijos koeficientas yra statistinis rodiklis, kuris rodo, kaip gerai statistinis modelis gali prognozuoti rezultatus. Šis koeficientas yra skaičius nuo 0 iki 1, kurio vertė parodo, kokia dalis variaci-

jos rezultate yra paaiškinama modelio. Kuo aukštesnis determinacijos koeficientas, tuo geriau modelis atitinka duomenis ir gali prognozuoti rezultatus. Šis koeficientas taip pat žinomais kaip R kvadratas [8].

Logistinė regresija

Logistinė regresija apibūdina santykius tarp kiekybinių ir kokybinių (kategoriniams) kintamųjų. Šio tipo statistinis modelis dažnai naudojamas klasifikavimui ir nuspėjamajai analizei. Logistinė regresija įvertina įvykio tikimybę, remiantis tam tikru nepriklausomų kintamųjų duomenų rinkiniu. Logistinės regresijos lygtis:

$$ln(\pi/(1-\pi)) = \beta_0 + \beta_1 * X_1 + \ldots + \beta_k * X_k,$$

kur π yra sėkmės tikimybė, $\beta_0, \beta_1, ..., \beta_k$ koeficientai, o $X_1, X_2, ..., X_k$ yra nepriklausomi kintamieji [7].

Atsitiktinis miškas

Atsitiktinis miškas (angl. Random forest) yra mašininio mokymosi algoritmas, kurį naudoja klasifikavimo ir regresijos uždaviniams spręsti. Jis sukuria kelis sprendimų medžius imdamas skirtingas testuojamų duomenų imtis ir pateikia daugiausiai kartų gautą rezultatą [9].

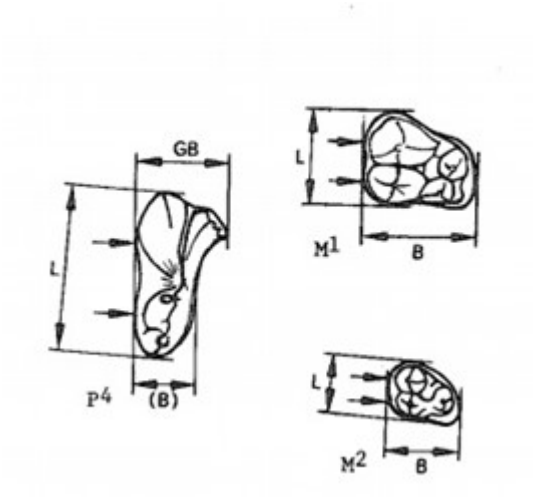
3 Tiriamoji dalis

3.1 Duomenų aprašymas

Šiam tyrimui buvo panaudotos Lietuvoje surinktų rudųjų lapių kaukolių kolekcija, kuri yra laikoma Kauno T. Ivanausko zoologijos muziejuje. Didžioji dalis kaukolių buvo gautos iš medžiotojų, o kai kurios iš jų buvo surinktos ekspedicijų metu, kai buvo rasti žuvusiu lapių skeletai. Iš viso buvo ištirta 230 kaukolių, iš kurių 113 buvo patinų ir 117 patelių. Matavimai, atlikti pagal A. von den Driesch (1976) metodiką, buvo vykdyti elektroniniu slankmačiu su 0,01 mm tikslumu. Tyrimui buvo naudojami viršutiniai ir apatiniai krūminiai dantys, kurie yra rekomenduojami tirti mėsėdžiams, kaip nurodoma šioje metodikoje [2]. P4 dantis yra prieškrūminis (lotyniškai Premolares), o M1, M2 ir M3 dantys – krūminiai (lotyniškai Molares).

Originaliame duomenų rinkinyje buvo duota kiekvieno danties ilgis ir plotis, žandikaulio ilgis bei tiriamojo objekto lytis. Tyrimo papildymui buvo įvesti nauji kintamieji, kurie aprašo danties plotą, bei santykį tarp danties ilgio ir pločio.

Viršutinio žandikaulio kintamieji:



1 pav.: Viršutinio žandikaulio krūminiai dantys [2].

P4_L – Didžiausias P4 danties ilgis.
P4_GB – Didžiausias P4 danties plotis.
P4_B – Mažiausias P4 danties plotis.
M1_L – Didžiausias M1 danties ilgis.
M1_B – Didžiausias M1 danties plotis.
M2_L – Didžiausias M2 danties ilgis.
M2_B – Didžiausias M2 danties plotis.
JL – Viršutinio žandikaulio ilgis.
P4_area – P4 danties plotas.
M1_area – M1 danties plotas.
M2_area – M2 danties plotas.
P4_L_to_GB_ratio – P4 danties ilgio ir didžiausio pločio santykis.

Didžiausias P4 danties ilgis padalintas iš didžiausias P4 danties pločio.

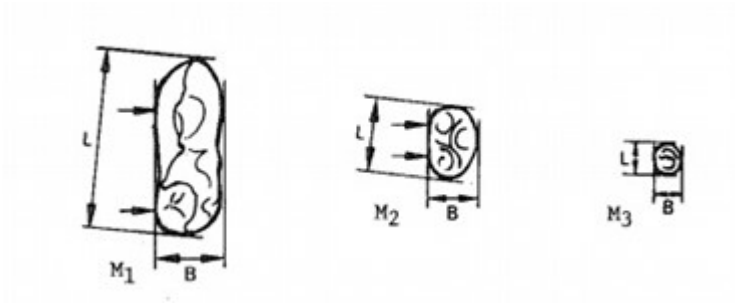
P4_L_to_B_ratio - P4 danties ilgio ir mažiausio pločio santykis. Didžiausias P4 danties ilgis padalintas iš mažiausio P4 danties pločio.

M1_L_to_B_ratio – M1 danties ilgio ir pločio santykis. Didžiausias M1 danties ilgis padalintas iš didžiausio M1 danties pločio.

M2_L_to_B_ratio – M2 danties ilgio ir pločio santykis. Didžiausias M2 danties ilgis padalintas iš didžiausio M2 danties pločio.

sex – Lytis.

Apatinio žandikaulio kintamieji:



2 pav.: Apatinio žandikaulio krūminiai dantys [2].

M1_L – Didžiausias M1 danties ilgis.

M1_B – Didžiausias M1 danties plotis.

M2_L – Didžiausias M2 danties ilgis.

M2_B – Didžiausias M2 danties plotis.

M3_L - Didžiausias M3 danties ilgis.

M3_B - Didžiausias M3 danties plotas.

JL – Apatinio žandikaulio ilgis.

M1_area – M1 danties plotas.

M2_area – M2 danties plotas.

M3_area – M3 danties plotas.

M1_L_to_B_ratio – M1 danties ilgio ir pločio santykis. Didžiausias M1 danties ilgis padalintas iš didžiausio M1 danties pločio.

M2_L_to_B_ratio – M2 danties ilgio ir pločio santykis. Didžiausias M2 danties ilgis padalintas iš didžiausio M2 danties pločio.

M3_L_to_B_ratio – M3 danties ilgio ir pločio santykis. Didžiausias M3 danties ilgis padalintas iš didžiausio M3 danties pločio.

sex – Lytis.

3.2 Aprašomoji statistika

	Vidurkis	Mediana	Standartinis nuokrypis	Maksimumas	Minimumas
P4_L	14.10	14.07	0.71	16.02	11.94
P4_GB	6.36	6.30	0.69	8.64	4.97
P4_B	5.05	5.05	0.34	6.05	4.12
M1_L	8.57	8.58	0.73	10.31	5.23
M1_B	11.27	11.23	0.68	13.34	9.33
M2_L	5.05	5.03	0.51	6.56	3.48
M2_B	8.10	8.13	0.62	9.85	5.31
JL	53.19	53.03	2.51	60.49	47.58
P4_area	80.60	80.27	8.93	113.15	57.73
M1_area	96.75	95.40	11.76	126.60	58.05
M2_area	41.00	40.75	5.90	58.58	24.43
P4_L_to_GB_ratio	2.24	2.22	0.23	2.94	1.60
P4_L_to_B_ratio	2.80	2.79	0.18	3.38	2.36
M1_L_to_B_ratio	0.76	0.77	0.06	0.91	0.47
M2_L_to_B_ratio	0.63	0.62	0.07	0.87	0.41

1 lentelė: Viršutinio žandikaulio kintamųjų aprašomoji statistika.

	Vidurkis	Mediana	Standartinis nuokrypis	Maksimumas	Minimumas
M1_L	15.08	15.10	0.82	17.50	12.31
M1_B	5.68	5.65	0.35	6.62	4.68
M2_L	6.88	6.95	0.63	8.30	4.59
M2_B	5.13	5.10	0.40	6.24	4.19
M3_L	3.15	3.20	0.47	4.41	1.63
M3_B	3.01	3.02	0.28	3.87	2.10
JL	59.28	59.37	2.64	67.12	53.72
M1_area	85.78	85.46	8.29	112.88	64.82
M2_area	35.36	35.24	4.98	50.35	23.42
M3_area	9.54	9.56	2.01	16.67	4.27
M1_L_to_GB_ratio	2.66	2.67	0.18	3.11	2.19
M2_L_to_B_ratio	1.35	1.36	0.13	1.60	0.88
M3_L_to_B_ratio	1.05	1.07	0.15	1.65	0.56

2 lentelė: Apatinio žandikaulio kintamųjų aprašomoji statistika.

Pirmojoje lentelėje pateikiama viršutinio žandikaulio kintamųjų statistika, antrojoje lentelėje – apatinio žandikaulio. Šiose lentelėse galime matyti kiekvieno kintamojo vidurkius, mediana, standartinius nuokrypius, maksimalią ir minimalią reikšmę.

3.3 Modelių sudarymas pasitelkiant Rattle paketą

Viršutinio ir apatinio žandikaulių krūminių dantų osteometrijos rezultatams buvo atlikta analizė naudojantis R programavimo kalbos Rattle paketu, kuris gali sukurti įvairius modelius pasitelkiant mašininį mokymąsi (angl. machine learning), grafiškai pavaizduoti turimus duomenis, atlikti testus, kurie yra skirti modelio patikimumui įvertinti, ir atlikti kitas funkcijas. Šiame darbe hipotezių tikrinimui reikšmingumo lygmuo $\alpha = 0.05$.

3.3.1 Viršutinio ir apatinio žandikaulių krūminių dantų osteometrijos rezultatų tiesinė regresija prognozuojant žandikaulio ilgį

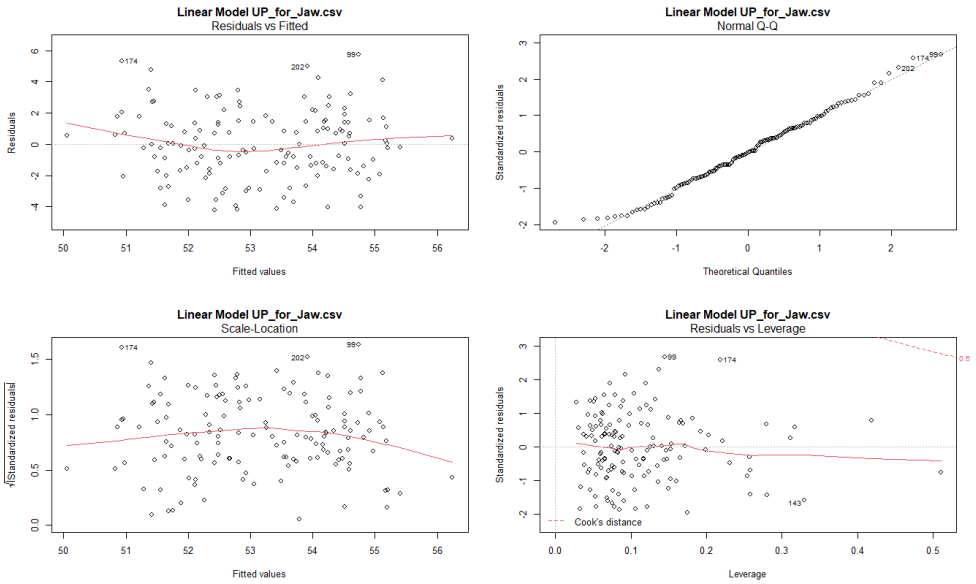
Šios analizės tikslas buvo sudaryti tiesinės regresijos modelį, kuris prognozuotų žandikaulio ilgį pagal turimus dantų matavimus. Taigi, naudojantis Rattle paketu buvo sudaryti du tiesinės regresijos modeliai viršutiniam ir apatiniam žandikauliams su visai turimais kintamaisiais ir šių modelių apibūdinimui buvo sudaryta po keturis grafikus, kurie padeda lengviau įvertinti turimus modelius.

„Residuals vs. Fitted" grafikas rodo stebėtų ir prognozuojamų reikšmių skirtumą (liekamoji paklaida) atsižvelgiant į prognozuotas reikšmes. Jis padeda įvertinti, ar stebėtų ir prognozuojamų reikšmių skirtumas turi struktūrą (dėsningumą) ir ar priklausomojo kintamojo santykis su nepriklausomaisiais kintamaisiais yra tiesinis. Šiame grafiniame paveiksle galima patikrinti, ar liekamoji paklaida yra atsitiktinai išbarsčiusi apie nulį be jokios aiškios struktūros [10].

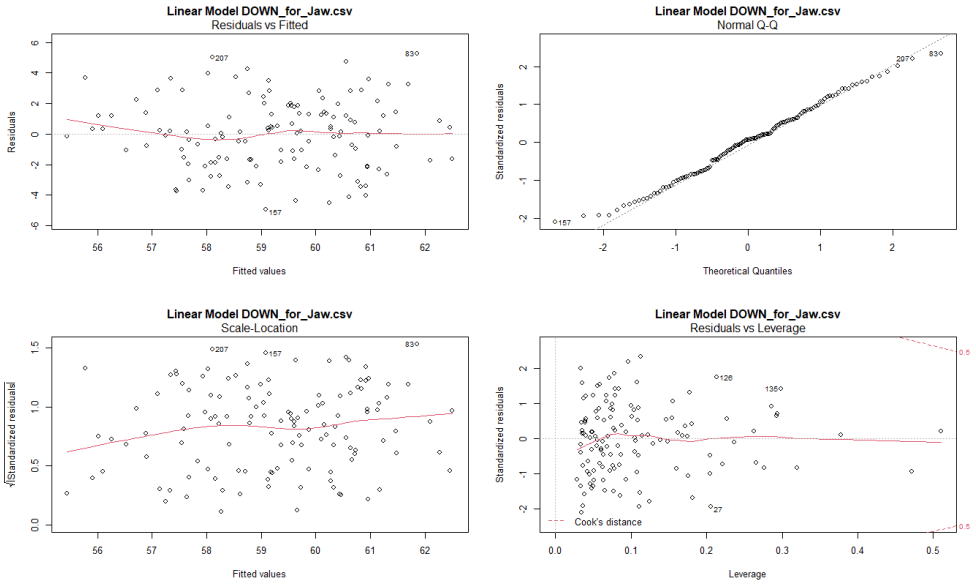
„Normal Q-Q" grafikas naudojamas įvertinti, ar liekamoji paklaida atitinka normalųjį pasiskirstymą. Jame yra pavaizduojami liekamosios paklaidos kvantiliai lyginant su kvantiliais iš standartinio normalaus pasiskirstymo. Jei liekamoji paklaida yra normaliai pasiskirsčiusi, taškai grafike turėtų maždaug sudaryti tiesę. Nuokrypiai nuo tiesės rodo nukrypimo nuo įprasto pasiskirstymo [11].

„Scale-Location" grafikas nagrinėja liekamosios paklaidos sklaidą ar dispersiją. Jame yra pavaizduota standartizuota liekamosios paklaidos (arba absoliučių verčių) kvadratinė šaknis atžvilgiu modelio prognozuojamoms reikšmės. Šis grafikas padeda patikrinti, ar liekamosios paklaidos kintamumas yra pastovus per prognozuojamų reikšmių intervalą. Idealioje situacijoje taškai turėtų būti atsitiktinai išbarstyti. [12].

„Residuals vs. Leverage" grafikas padeda nustatyti įtaką turinčius duomenų taškus, kurie turi didelį svertą (angl. leverage) regresijos modeliui. Jis pavaizduoja standartizuotas liekamąsias paklaidas pagal sverto reikšmes, kurios matuoja, kiek nepriklausomų kintamųjų stebėjimo vertė skiriasi nuo vidutinės vertės. Taškai, kurie turi didelį svertą ir didelę liekamąją paklaidą, gali stipriai įtakoti regresijos rezultatus [13].



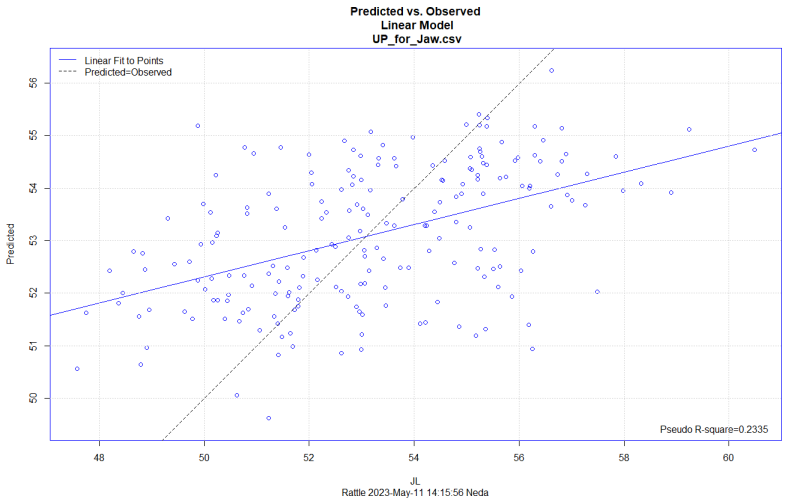
3 pav.: Viršutinio žandikaulio krūminių dantų osteometrijos rezultatų tiesinės regresijos grafikai.



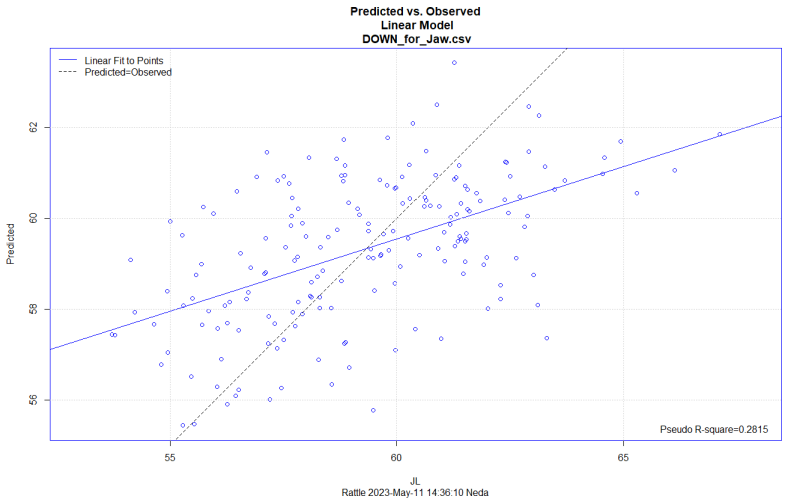
4 pav.: Apatinio žandikaulio krūminių dantų osteometrijos rezultatų tiesinės regresijos grafikai.

Tiek viršutinio tiek apatinio žandikaulio krūminių dantų osteometrijos rezultatų tiesinės regresijos žandikaulio ilgiui prognozuoti „Residuals vs. Fitted" grafikai parodo, liekamosios paklaidos yra atsitiktinai išbarsčiusios apie nulį be jokios aiškios struktūros. „Normal Q-Q" grafikai rodo, kad nėra prieštaravimo normaliajam pasiskirtymui. „Scale-Location" grafikai rodo, kad liekamosios paklaidos atsitiktinai išbarstytos. Taip pat „Residuals vs. Leverage" grafikai išskyrė po tris taškus, kurie turi didelį svertą ir didelę liekamąją paklaidą. Modelių patikimumui įvertinti buvo sudaryti „Predicted vs. observed" grafikai.

„Predicted vs. observed" grafike rodoma numatoma vertė palyginta su faktine kiekvieno stebėjimo verte testuojamų duomenų rinkinyje. Mėlyna linija vaizduoja nubrėžtus taškus, o pilka linija rodo, kur taškai išsidėstytų, jei visos numatytos vertės puikiai atitiktų stebimas. Kuo arčiau dvi linijos, tuo geresnis modelio veikimas [14].



5 pav.: Viršutinio žandikaulio krūminių dantų osteometrijos rezultatų tiesinės regresijos „Predicted vs. observed" grafikas.



6 pav.: Apatinio žandikaulio krūminių dantų osteometrijos rezultatų tiesinės regresijos „Predicted vs. observed" grafikas.

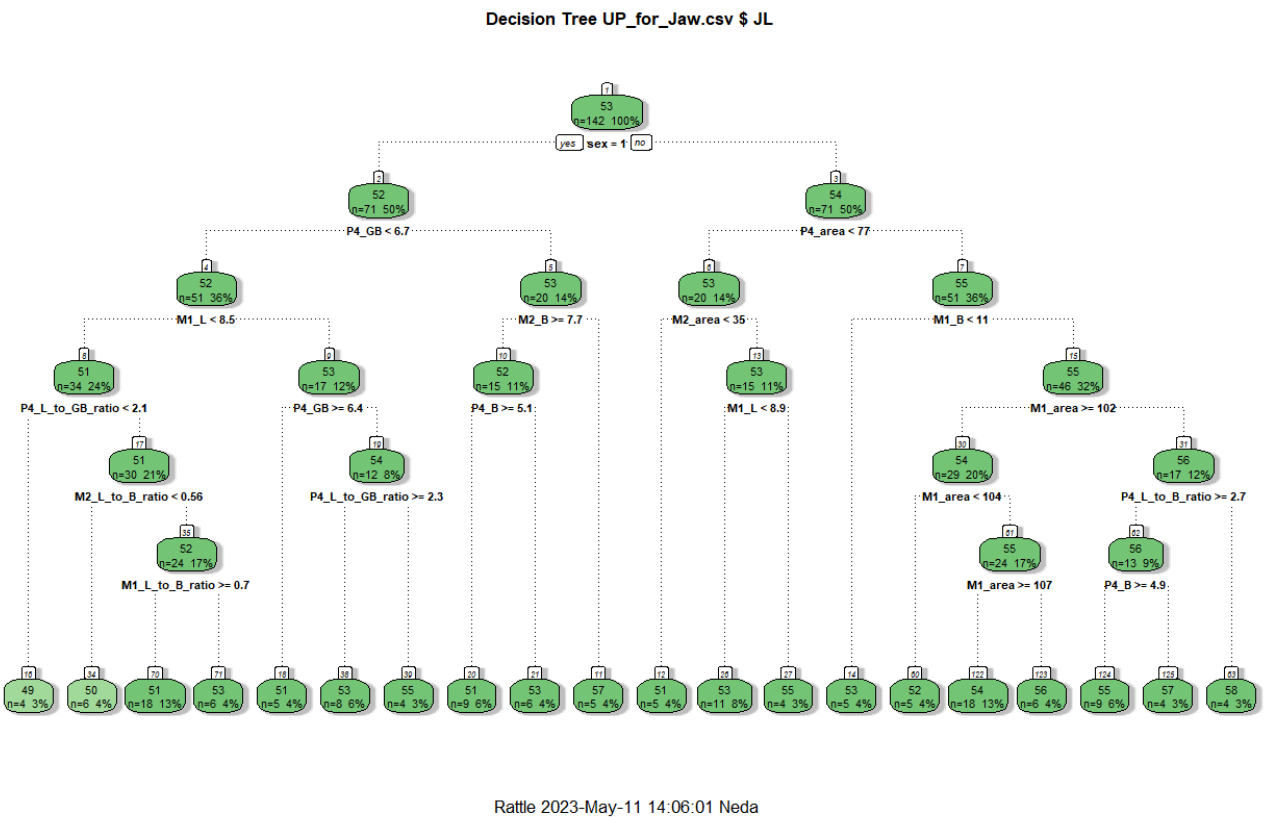
R kvadrato reikšmė, dar žinoma kaip determinacijos koeficientas, matuoja priklausomo kintamojo dispersijos proporciją, kurią galima paaiškinti nepriklausomais kintamaisiais tiesinės regresijos modelyje. Viršutinio žandikaulio modelyje determinacijos koeficientas yra 0.23, o apatinio žandikaulio modelyje determinacijos koeficientas lygus 0.28. Sprendžiant iš R kvadrato reikšmių abu šie modeliai nėra tinkami.

Deja, modeliai susiduria su multikolinearumo problema.

Taip pat buvo sudaryti tiesinės regresijos modeliai su pradiniais duomenimis, kuriuose nebuvo įtraukta dantų ploto ar santykio tarp danties ilgio ir pločio. Abiems tiesinės regresijos modeliams su pradiniais duomenimis „Residuals vs. Fitted" grafikai (19 pav. ir 20 pav.) parodo, liekamosios paklaidos yra atsitiktinai išbarsčiusios apie nulį be jokios aiškios struktūros. „Normal Q-Q" grafikai (19 pav. ir 20 pav.) rodo, kad nėra prieštaravimo normaliajam pasiskirtymui. „Scale-Location" grafikai (19 pav. ir 20 pav.) rodo, kad liekamosios paklaidos atsitiktinai išbarstytos. Taip pat „Residuals vs. Leverage" (19 pav. ir 20 pav.) grafikai išskyrė po tris taškus, kurie turi didelį svertą ir didelę liekamąją paklaidą. Šiems modeliams nėra multikolinearumo problemos. „Predicted vs. observed" grafikai (21 pav. ir 22 pav.) rodo, kad viršutinio žandikaulio modelyje R kvadratas lygus 0.23, o apatinio žandikaulio modelyje R kvadratas - 0.28. Abu modeliai nėra tinkami ir R kvadato reikšmės mažai skiriasi nuo tiesinės regresijos modelių su naujais kintamaisiais.

3.3.2 Viršutinio ir apatinio žandikaulių krūminių dantų osteometrijos rezultatų sprendimų medžiai prognozuojant žandikaulio ilgį

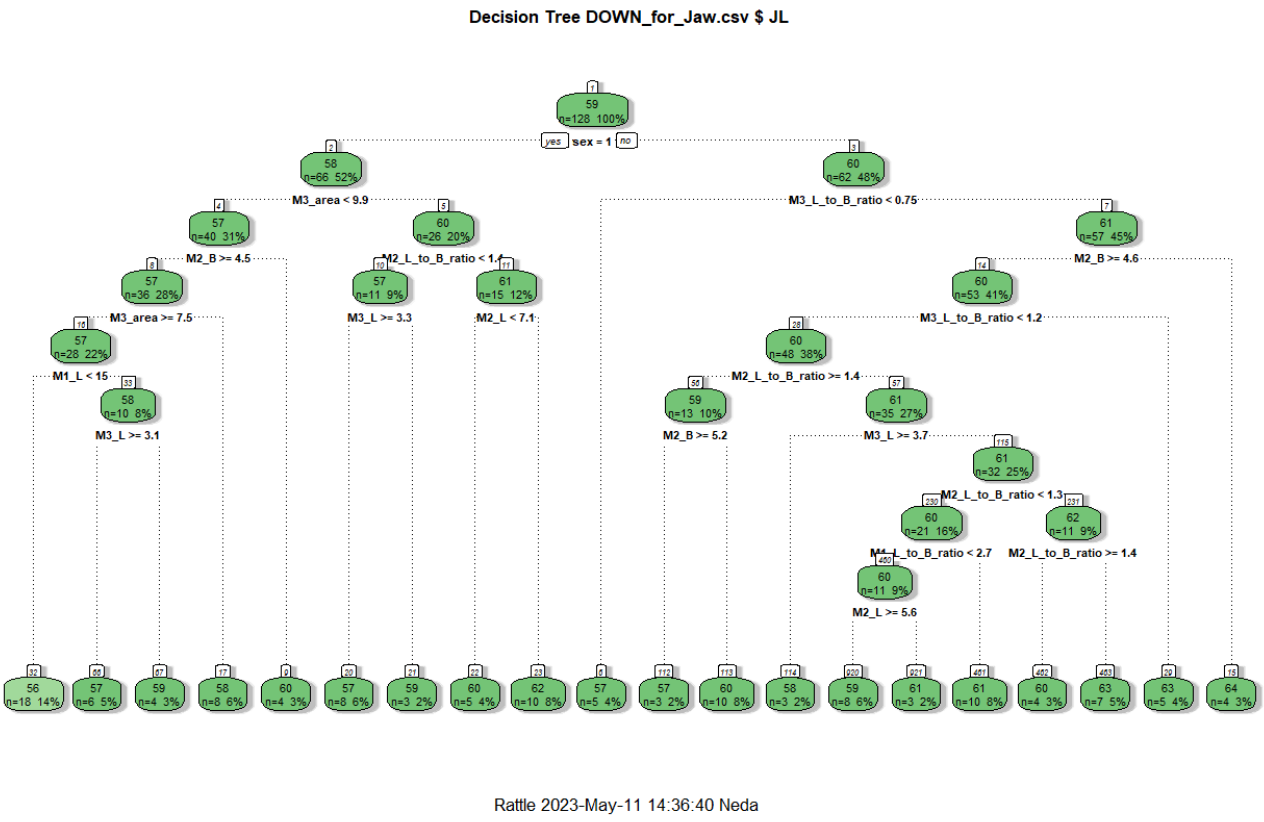
Šios analizės tikslas buvo sudaryti sprendimų medį, kuris prognozuotų žandikaulio ilgį pagal turimus dantų matavimus. Sprendimų medžiams apatiniam ir viršutiniam žandikauliams sudaryti buvo naudojamas Rattle paketas.



7 pav.: Viršutinio žandikaulio krūminių dantų osteometrijos rezultatų sprendimų medis.

Medžio pradžia yra šaknis, o nuo jo, priklausomai nuo galimų variantų, sprendimai skaidomi į šakas. Kiekviena šakos viršūnė sprendimų medyje yra pasirinkimas tarp galimų alternatyvų, o kiekvienas lapas atitinka konkretų sprendimą. Viršutinio žandikaulio krūminių dantų osteometrijos rezultatų sprendimų medžio šaknis turi 142 tiriamuosius objektus, šios imties vidutinis žandikaulio ilgis yra 53 mm ir šaknyje yra 100 procentų imties. Sprendimų medis toliau imtį skirto pagal lytį. Kairioji šaka atspindi pateles, kurių vidutinis žandikaulių ilgis yra 52 mm ir šią imtį sudaro 71 objektas, kas atitinka 50 procentų šaknies tiriamųjų objektų. Dešinioji šaka atspindi patinus, kurių vidutinis žandikaulių ilgis yra 54 mm ir šią imtį taip pat sudaro 71 objektas (50 procentų šaknies tiriamųjų objektų). Taip sprendimai skaidomi į tolimesnes šakas.

Šis sprendimų medis naudoja 13 kintamųjų: M1_area (M1 danties plotas), M1_B (didžiausias M1 danties plotis), M1_L (didžiausias M1 danties ilgis), M1_L_to_B_ratio (M1 danties ilgio ir pločio santykis), M2_area (M2 danties plotas), M2_B (didžiausias M2 danties plotis), M2_L_to_B_ratio (M2 danties ilgio ir pločio santykis), P4_area (P4 danties plotas), P4_B (mažiausias P4 danties plotis), P4_GB (didžiausias P4 danties plotis), P4_L_to_B_ratio (P4 danties ilgio ir mažiausio pločio santykis), P4_L_to_GB_ratio (P4 danties ilgio ir didžiausio pločio santykis), sex (lytis). Šis sprendimų medis turi 20 skirtingų baigčių. Matome, kad sprendimų medis naudoja septynis naujai įvestus kintamuosius: M1_area, M1_L_to_B_ratio, M2_area, M2_L_to_B_ratio, P4_area, P4_L_to_B_ratio, P4_L_to_GB_ratio.

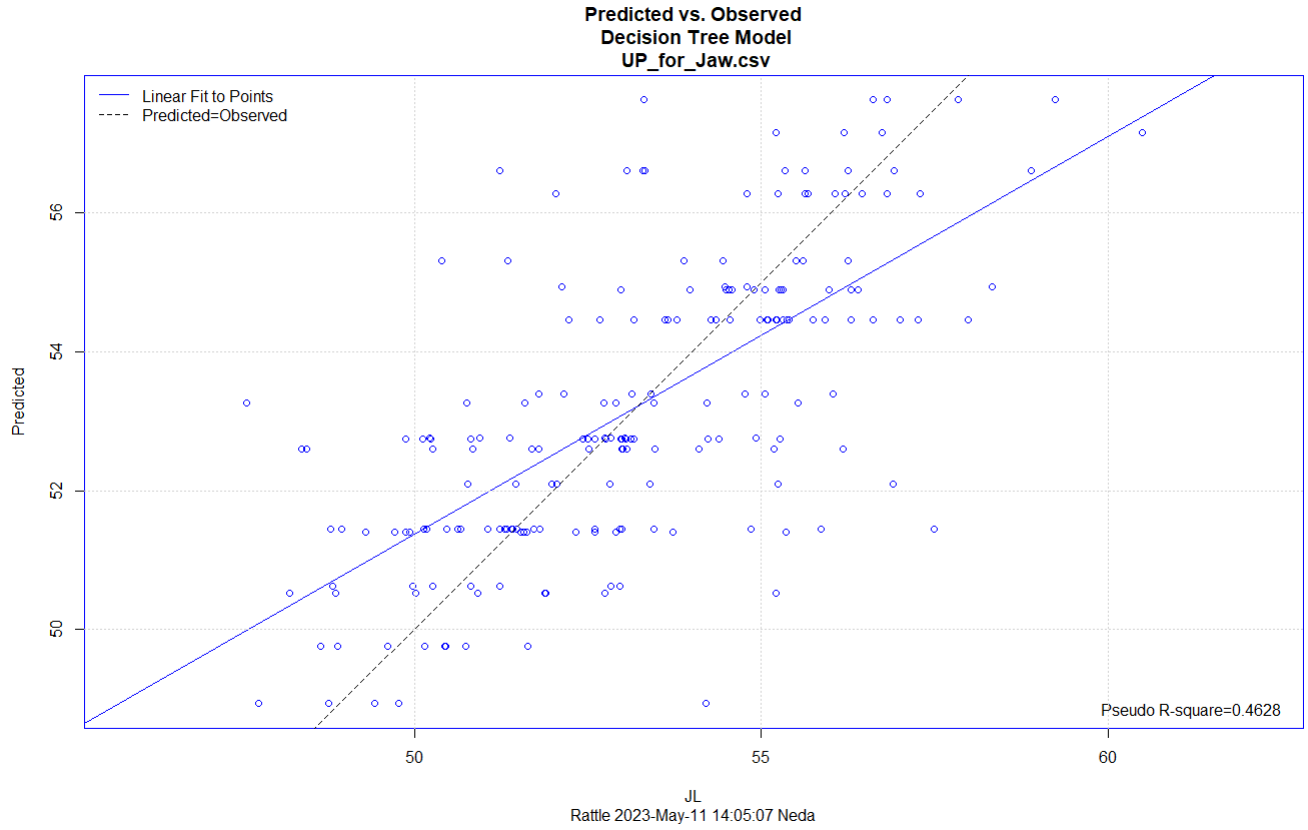


8 pav.: Apatinio žandikaulio krūminių dantų osteometrijos rezultatų sprendimų medis.

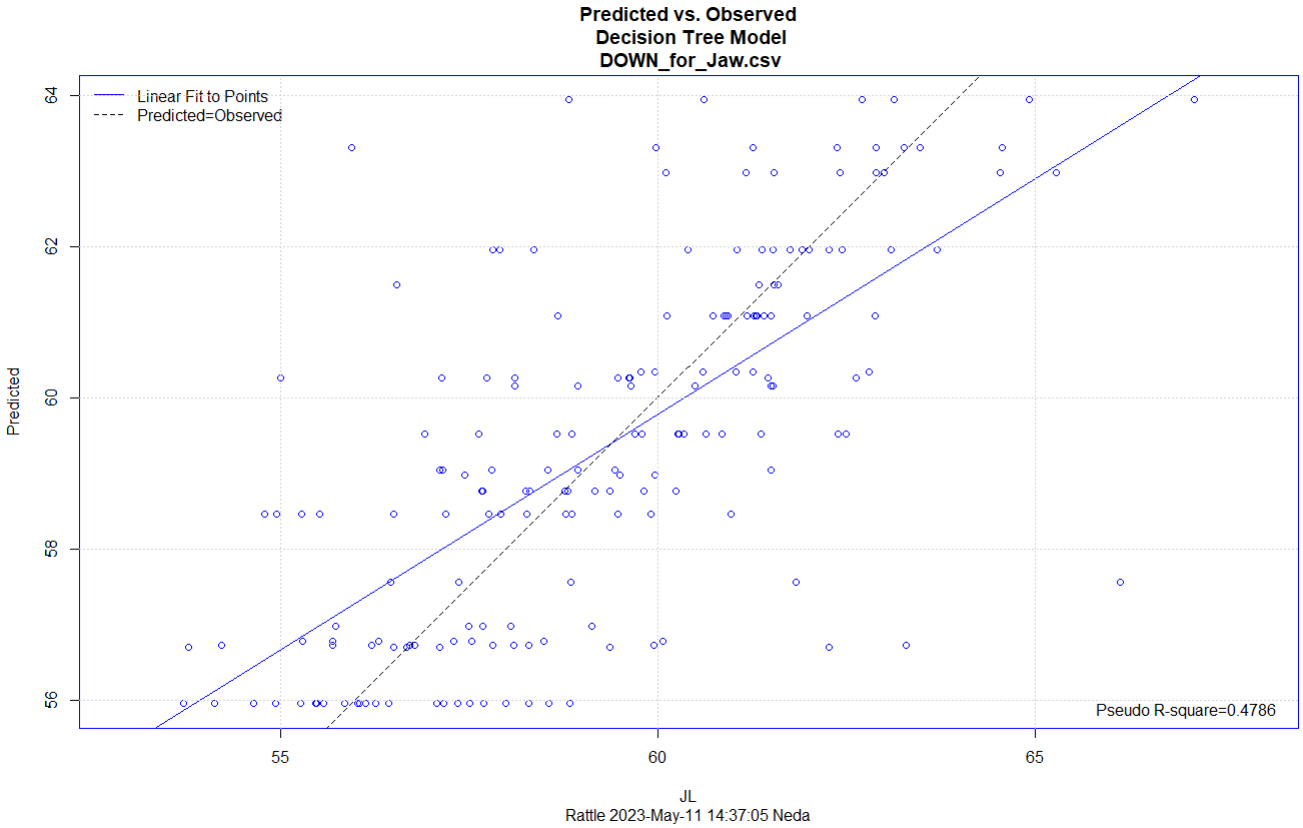
Apatinio žandikaulio krūminių dantų osteometrijos rezultatų sprendimų medžio šaknis turi 128

objektus, šios imties vidutinis žandikaulio ilgis yra 59 mm ir šaknyje yra 100 procentų imties. Sprendimų medis toliau imtį skirto pagal lytį. Kairioji šaka atspindi pateles, kurių vidutinis žandikaulių ilgis yra 58 mm ir šią imtį sudaro 66 objektai, kas atitinka 52 procentus šaknies tiriamųjų objektų. Dešinioji šaka atspindi patinus, kurių vidutinis žandikaulių ilgis yra 60 mm ir šią imtį sudaro 62 objektai, kas yra 48 procentai šaknies tiriamųjų objektų. Taip sprendimai skaidomi į tolimesnes šakas.

Šis sprendimų medis naudoja 9 kintamuosius: M1_L (didžiausias M1 danties ilgis), M1_L_to_B_ratio (M1 danties ilgio ir pločio santykis), M2_L (didžiausias M2 danties ilgis), M2_B (didžiausias M2 danties plotis), M3_area (M3 danties plotas), M2_L_to_B_ratio (M2 danties ilgio ir pločio santykis), M3_L (didžiausias M3 danties ilgis), M3_L_to_B_ratio (M3 danties ilgio ir pločio santykis), sex (lytis). Šis sprendimų medis taip pat turi 20 skirtingų baigčių. Galime pastebėti, jog sprendimų medis naudoja keturis naujai įvestus kintamuosius: M1_L_to_B_ratio, M3_area, M2_L_to_B_ratio, M3_L_to_B_ratio.



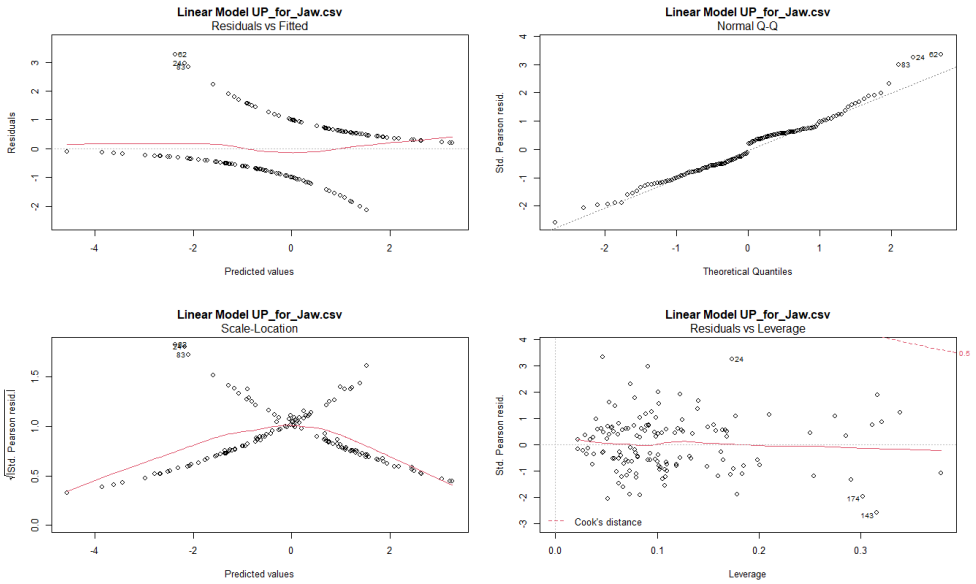
9 pav.: Viršutinio žandikaulio krūminių dantų osteometrijos rezultatų sprendimų medžio „Predicted vs. observed" grafikas.



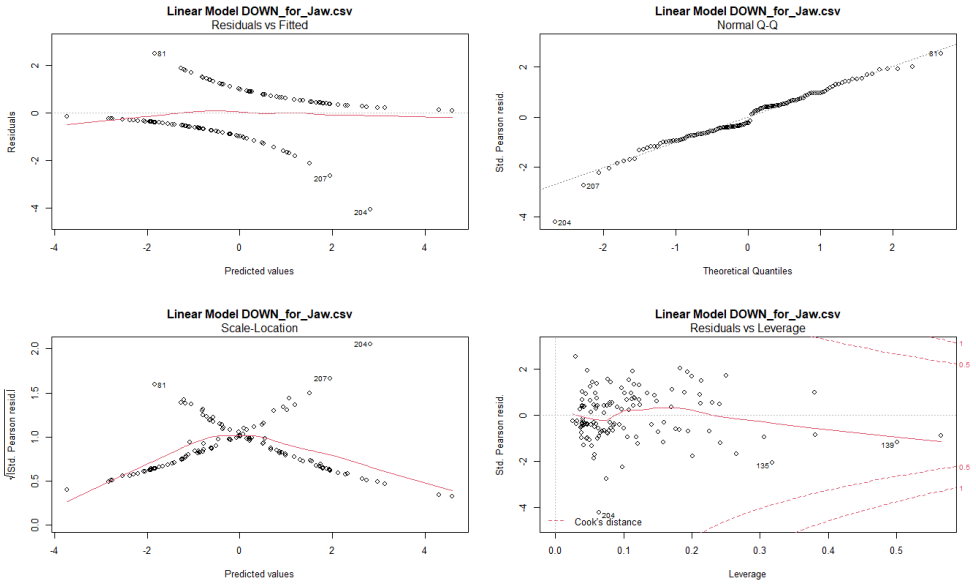
10 pav.: Apatinio žandikaulio krūminių dantų osteometrijos rezultatų sprendimų medžio „Predicted vs. observed" grafikas.

Viršutinio žandikaulio modelyje determinacijos koeficientas yra 0.46, o apatinio žandikaulio modelyje determinacijos koeficientas lygus 0.48, kas reikštų, jog modeliai gana tiksliai aprašo duomenis.

3.3.3 Viršutinio ir apatinio žandikaulių krūminių dantų osteometrijos rezultatų logistinė regresija prognozuojant lapės lytį



11 pav.: Viršutinio žandikaulio krūminių dantų osteometrijos rezultatų logistinės regresijos grafikai.



12 pav.: Apatinio žandikaulio krūminių dantų osteometrijos rezultatų logistinės regresijos grafikai.

Viršutinio ir apatinio žandikaulio krūminių dantų osteometrijos rezultatų logistinės regresijos lyčiai prognozuoti „Residuals vs. Fitted" grafikai parodo, liekamosios paklaidos nėra atsitiktinai išbarsčiusios apie nulį, kas sako, jog logistinė regresija nėra geriausias modelis turimiems duomenims. „Normal Q-Q" grafikai yra šiek tiek nukrypę nuo normalaus pasiskirstymo, kas yra pakankamai dažnas įvykis logistinėje regresijoje. „Scale-Location" grafikai rodo, kad liekamosios paklaidos nėra atsitiktinai išbarstytos, kas indikuoja heteroskedastiškumą. Taip pat „Residuals vs. Leverage" grafikai išskyrė po tris taškus, kurie turi didelį svertą ir didelę liekamąją paklaidą.

Rattle pakete esanti klaidų matrica (klasifikavimo lentelė) apibūdina klasifikavimo modelio veikimą, lyginant numatomas ir faktines klases. Lentelėje eilutės žymi faktines klases, o stulpeliai – numatomas. Ši matrica turi dvi klases pagal lytį. „F“ (patelė) ir „M“ (patinas). Lentelėje esantys skaičiai rodo atvejų, patenkančių į kiekvieną kategoriją, procentinę dalį pagal modelio prognozes. Klaidų matrica suteikia įžvalgų apie modelio tikslumą ir leidžia įvertinti jo veikimą klasifikuojant atvejus pagal numatomas ir faktines vertes.

	Numatoma F (proc.)	Numatoma M (proc.)	Klaida (proc.)
Faktinė F (proc.)	35	15.3	30.4
Faktinė M (proc.)	12.8	36.9	25.7

3 lentelė: Klaidų matrica viršutinio žandikaulio krūminių dantų osteometrijos rezultatų tiesinei regresijai.

Vertė 35 viršutiniame kairiajame langelyje rodo, kad modelis teisingai numatė 35 procentų atvejų, kurie iš tikrųjų buvo patelės. Vertė 15.3 viršutiniame dešiniajame langelyje rodo, kad modelis neteisingai numatė 15.3 procentų atvejų kaip „M" (patinas), kai jie iš tikrųjų buvo „F" (patelės). Vertė 30.4 apatiniame rodo bendrą „F“ (patelių) klasės klaidų lygį.

Vertė 12.8 viršutiniame kairiajame langelyje rodo, kad modelis teisingai numatė 12.8 procentų atvejų, kurie iš tikrųjų buvo patinai. Vertė 36.9 viršutiniame dešiniajame langelyje rodo, kad modelis neteisingai numatė 36.9 procentų atvejų kaip „F" (patelės), kai jie iš tikrųjų buvo „M" (patinai). Vertė 25.7 apatiniame rodo bendrą „M“ (patinų) klasės klaidų lygį.

Vidutinė klasės klaida yra 28.05 procentai, o bendras modelio klaidingumas yra 28.10 procentai.

	Numatoma F (proc.)	Numatoma M (proc.)	Klaida (proc.)
Faktinė F (proc.)	38.8	12.6	24.5
Faktinė M (proc.)	14.2	34.4	29.2

4 lentelė: Klaidų matrica apatinio žandikaulio krūminių dantų osteometrijos rezultatų tiesinei regresijai.

Vertė 38.8 viršutiniame kairiajame langelyje rodo, kad modelis teisingai numatė 38.8 procentų atvejų, kurie iš tikrųjų buvo patelės. Vertė 12.6 viršutiniame dešiniajame langelyje rodo, kad modelis neteisingai numatė 12.6 procentų atvejų kaip „M" (patinas), kai jie iš tikrųjų buvo „F" (patelės). Vertė 24.5 apatiniame rodo bendrą „F“ (patelių) klasės klaidų lygį.

Vertė 14.2 viršutiniame kairiajame langelyje rodo, kad modelis teisingai numatė 14.2 procentų atvejų, kurie iš tikrųjų buvo patinai. Vertė 34.4 viršutiniame dešiniajame langelyje rodo, kad modelis neteisingai numatė 34.4 procentų atvejų kaip „F" (patelės), kai jie iš tikrųjų buvo „M" (patinai).

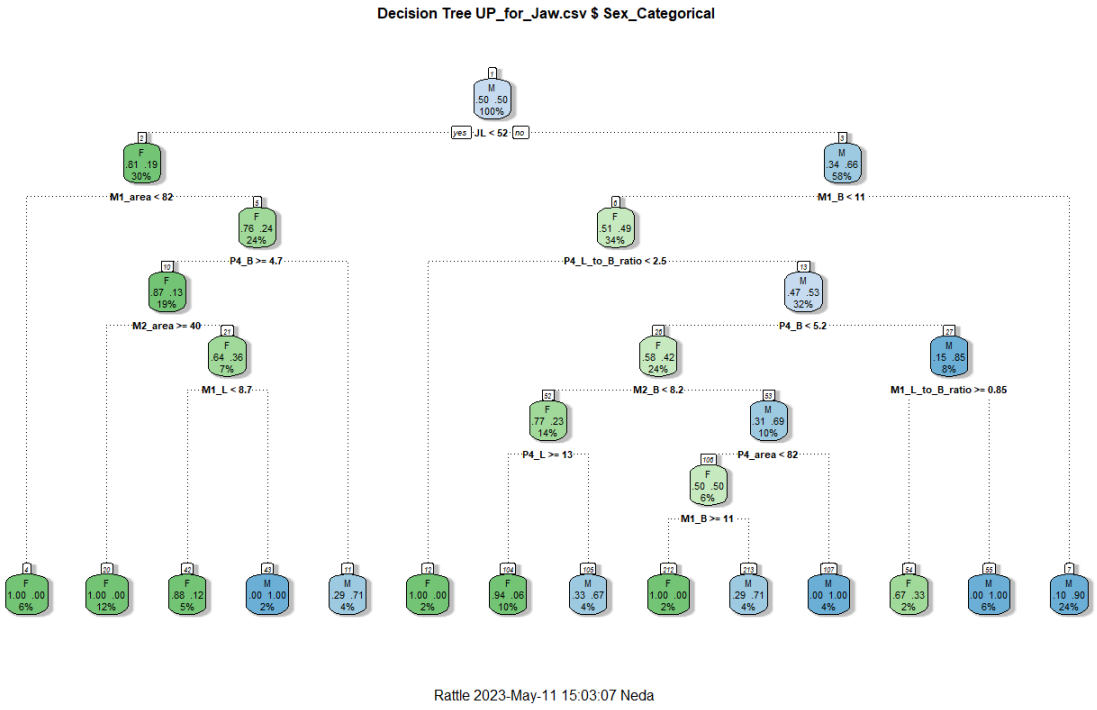
Vertė 29.2 apatiniame rodo bendrą „M“ (patinų) klasės klaidų lygį.

Vidutinė klasės klaida yra 26.85 procentai, o bendras modelio klaidingumas yra 26.80 procentai.

Deja, modeliai susiduria su multikolinearumo problema.

Palyginimui buvo sudaryti logistinės regresijos modeliai su pradiniais duomenimis, kuriuose nebuvo įtraukta dantų ploto ar santykio tarp danties ilgio ir pločio. Viršutinio ir apatinio žandikaulio krūminių dantų osteometrijos rezultatų logistinės regresijos lyčiai prognozuoti „Residuals vs. Fitted" (23 pav. ir 24 pav.) grafikai parodo, liekamosios paklaidos nėra atsitiktinai išbarsčiusios apie nulį. „Normal Q-Q" (23 pav. ir 24 pav.) grafikai yra šiek tiek nukrypę nuo normalaus pasiskirstymo. „Scale-Location" (23 pav. ir 24 pav.) grafikai rodo, kad liekamosios paklaidos nėra atsitiktinai išbarstytos, kas indikuoja heteroskedastiškumą. Taip pat „Residuals vs. Leverage" (23 pav. ir 24 pav.) grafikai išskyrė po tris taškus, kurie turi didelį svertą ir didelę liekamąją paklaidą. Šiems modeliams nėra multikolinearumo problemos. Modelių tikslumui parodyti, buvo sudarytos klaidų matricos (5 lentelė ir 6 lentelė).Vidutinė klasės klaida viršutinio žandikaulio modelio su pradiniais kintamaisiais yra 26.60 procentai, apatinio - 27.40 procentai, o bendras modelio klaidingumas yra 26.60 procentai, apatinio 27.30 procentai. Šių modelių reikšmės mažai skiriasi nuo logistinės regresijos modelių su naujais kintamaisiais.

3.3.4 Viršutinio ir apatinio žandikaulių krūminių dantų osteometrijos rezultatų sprendimų medžiai prognozuojant lapės lytį

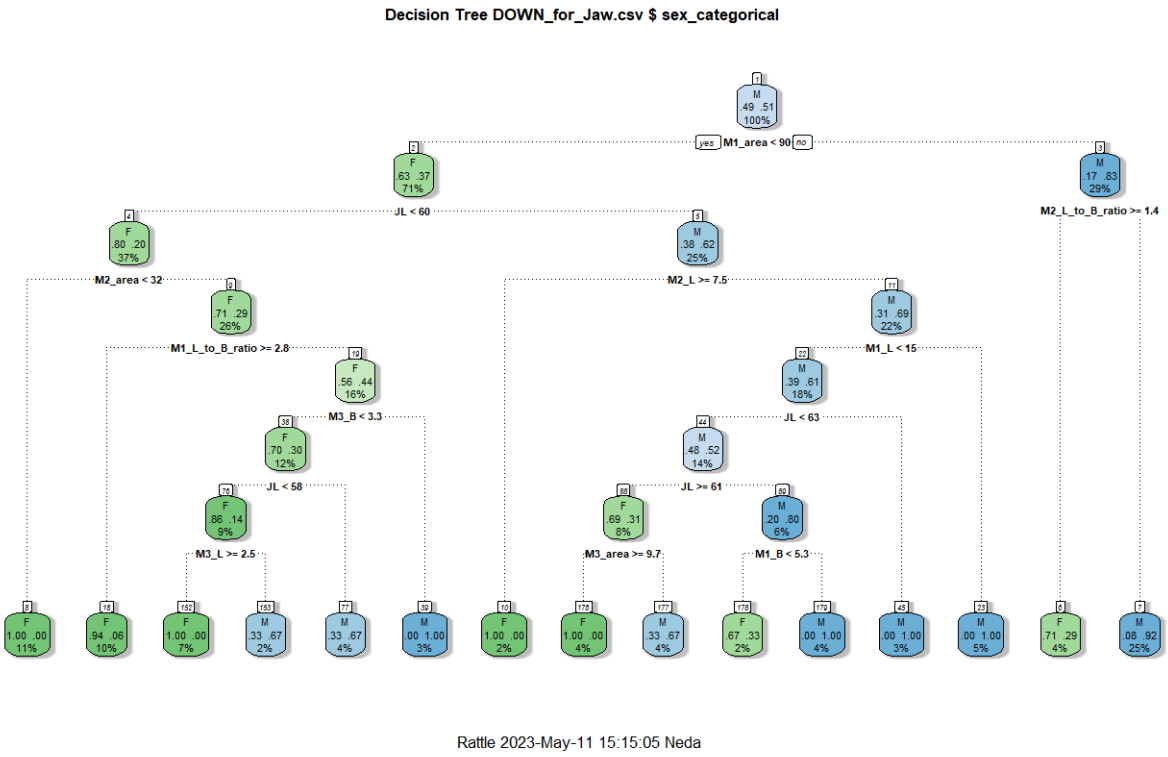


13 pav.: Viršutinio žandikaulio krūminių dantų osteometrijos rezultatų sprendimų medis.

Viršutinio žandikaulio krūminių dantų osteometrijos rezultatų sprendimų medžio, skirto lyčiai prognozuoti, šaknis turi 161 objektus, šioje imtyje vidutiniškai daugiau patinų. Sprendimų medis toliau imtį

skirto pagal žandikaulio ilgį. Kairioji šaka atspindi lapes, kurių žandikaulių ilgis yra mažesnis nei 52 mm ir joje vidutiniškai yra daugiau patelių. Dešinioji šaka atspindi lapes, kurių žandikaulių ilgis yra didesnis nei 52 mm ir šioje imtyje vidutiniškai daugiau patinų. Taip sprendimai skaidomi į tolimesnes šakas.

Šis sprendimų medis naudoja 11 kintamųjų: JL (žandikaulio ilgis), M1_L (didžiausias M1 danties ilgis), M1_B (didžiausias M1 danties plotis), M1_area (M1 danties plotas), M1_L_to_B_ratio (M1 danties ilgio ir pločio santykis), M2_B (didžiausias M2 danties plotis), M2_area (M2 danties plotas), P4_area (P4 danties plotas), P4_B (Mažiausias P4 danties plotis), P4_L (didžiausias P4 danties ilgis), P4_L_to_B_ratio (P4 danties ilgio ir mažiausio pločio santykis). Šis sprendimų medis turi 14 skirtingų baigčių, iš kurių septynios kategorizuoja į pateles ir septynios į patinus. Galime pastebėti, jog sprendimų medis naudoja net penkis naujai įvestus kintamuosius: M1_area, M1_L_to_B_ratio, M2_area, P4_area, P4_L_to_B_ratio.

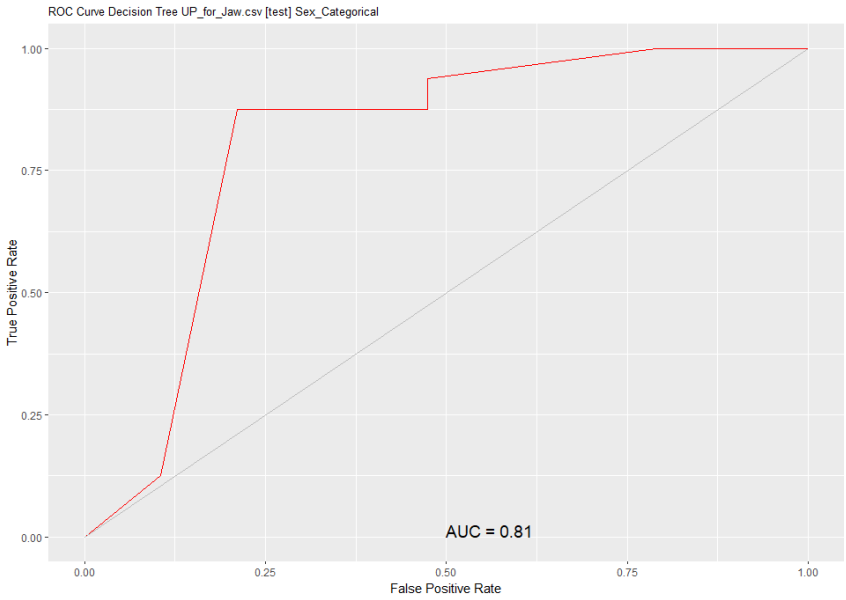


14 pav.: Apatinio žandikaulio krūminių dantų osteometrijos rezultatų sprendimų medis.

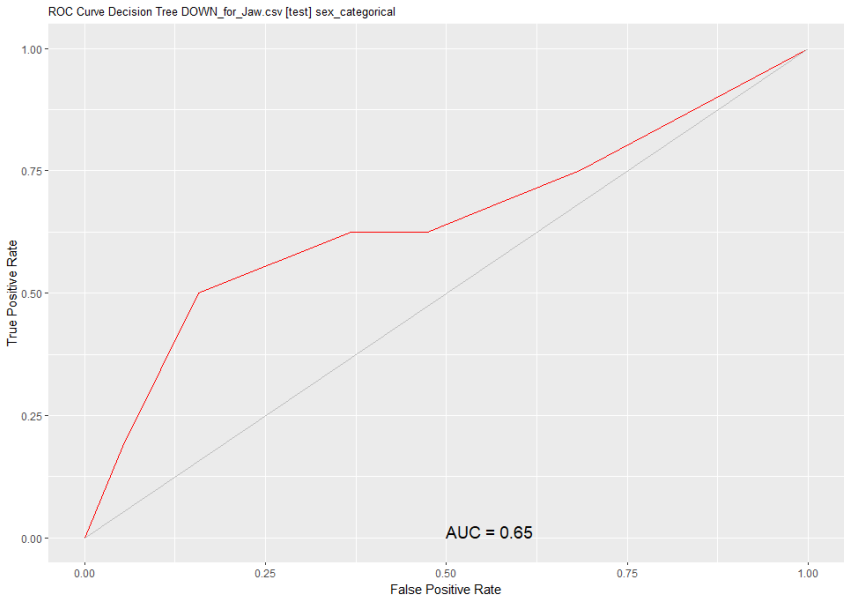
Apatinio žandikaulio krūminių dantų osteometrijos rezultatų sprendimų medžio, skirto lyčiai prognozuoti, šaknis turi 160 objektus, šioje imtyje vidutiniškai daugiau patinų. Sprendimų medis toliau imtį skirto pagal M1 danties plotą. Kairioji šaka atspindi lapes, kurių M1 danties plotas yra mažesnis nei 90 mm2 ir joje vidutiniškai yra daugiau patelių. Dešinioji šaka atspindi lapes, kurių M1 danties plotas yra didesnis nei 90 mm2 ir šioje imtyje vidutiniškai daugiau patinų. Taip sprendimai skaidomi į tolimesnes šakas.

Šis sprendimų medis taip pat naudoja 11 kintamųjų: JL (žandikaulio ilgis), M1_area (M1 danties plotas), M1_L (didžiausias M1 danties ilgis), M1_B (didžiausias M1 danties plotis), M1_L_to_B_ratio

(M1 danties ilgio ir pločio santykis), M2_area (M2 danties plotas), M2_L (didžiausias M2 danties ilgis), M2_L_to_B_ratio (M2 danties ilgio ir pločio santykis), M3_area (M3 danties plotas), M3_L (didžiausias M3 danties ilgis), M3_B (didžiausias M3 danties plotis). Šis sprendimų medis turi 15 skirtingų baigčių iš kurių septynios kategorizuoja į pateles ir aštuonios į patinus. Matome, kad sprendimų medis taip pat naudoja penkis naujai įvestus kintamuosius: M1_area, M1_L_to_B_ratio, M2_area, M2_L_to_B_ratio, M3_area.



15 pav.: Viršutinio žandikaulio krūminių dantų osteometrijos rezultatų sprendimų medžio ROC kreivė.



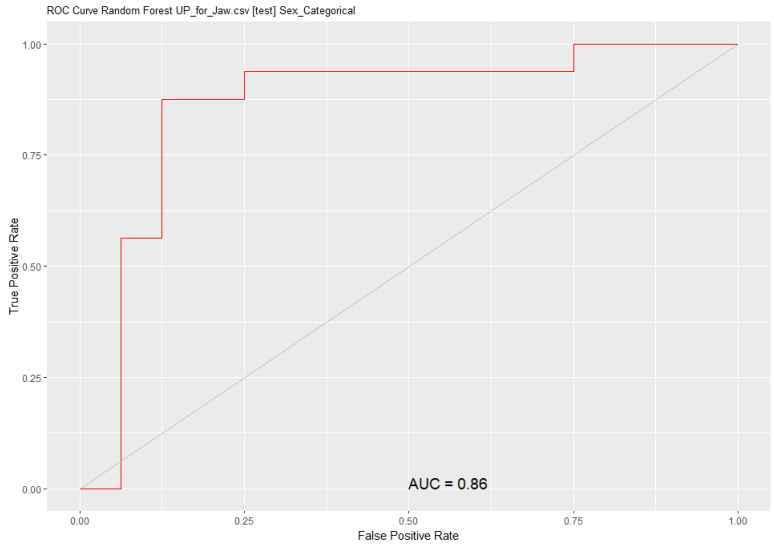
16 pav.: Apatinio žandikaulio krūminių dantų osteometrijos rezultatų sprendimų medžio ROC kreivė.

Ploto po viršutinio žandikaulio krūminių dantų osteometrijos rezultatų sprendimų medžio ROC kreive AUC yra lygi 0.81, kas indikuoja gerą modelio veikimą. Ploto po apatinio žandikaulio ROC kreive AUC yra 0.65, kas rodo silpną modelio veikimą.

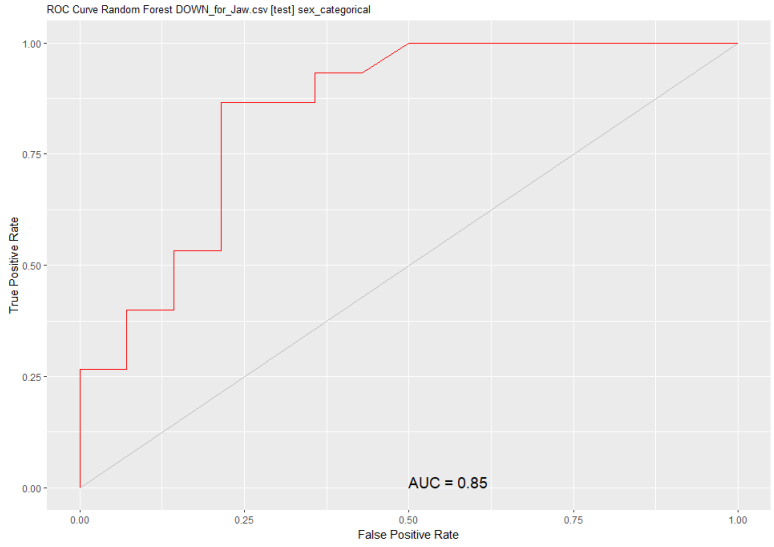
3.3.5 Viršutinio ir apatinio žandikaulių krūminių dantų osteometrijos rezultatų atsitiktinis miškas prognozuojant lapės lytį

Šios analizės tikslas buvo sudaryti atsitiktinį mišką, kuris prognozuotų tiriamojo objekto lytį pagal turimus dantų matavimus naudojant Rattle paketą. Tiek viršutinio, tiek apatinio žandikaulio atsitiktinių miškų sudarymui buvo naudojami visi jų kintamieji. Viršutinio žandikaulio 15, o apatinio – 13.

Atsitiktinio miško taikymas yra pranašesnis lyginant su sprendimų medžiais, nes yra tikslesni. Atsitiktinis miškas sudaro ne vieną, o kelis medžius ir sujungia jų prognozes, kas lemia, jog prognozė bus tikslesnė. Taip pat atsitiktinis miškas geriau susitvarko su didesniu duomenų kiekiu, bei yra mažiau jautrūs išskirtims.



17 pav.: Viršutinio žandikaulio krūminių dantų osteometrijos rezultatų atsitiktinio miško ROC kreivė.



18 pav.: Apatinio žandikaulio krūminių dantų osteometrijos rezultatų atsitiktinio miško ROC kreivė.

Ploto po viršutinio žandikaulio krūminių dantų osteometrijos rezultatų atsitiktinio miško ROC kreive AUC yra lygus 0.86, o apatinio žandikaulio AUC yra 0.85. Tiek apatinio, tiek viršutinio žandikaulių AUC reikšmės indikuoja puikų modelio veikimą.

4 Išvados

Šiame darbe buvo padarytos prognozės nustatyti lapių lytį ar jų žandikaulių ilgį remiantis turimais osteometrijos rezultatais. Prognozės buvo atliktos atskirai viršutiniam ir apatiniam lapės žandikauliui. Šioms prognozėms atlikti buvo naudojama tiesinė regresija, logistinė regresija, sprendimų medis ir atsitiktinis miškas. Modeliai buvo sudarinėjami naudojantis R programavimo kalba, „Rstudio" sistemos Rattle paketu.

Sudarius tiesinės regresijos modelius skirtus prognozuoti žandikaulio ilgius, pastebėta, jog modeliai susiduria su multikolinearumo problema ir R kvadrato reikšmės rodo, jog modeliai nėra tinkami. Su tomis pačiomis problemomis susiduria ir logistinės regresijos modeliai, kurie skirti prognozuoti lapės lytį. Sprendimų medžiai, kurie skirti prognozuoti tiek tiriamųjų objektų lytį, tiek žandikaulio ilgį, parodė žymiai geresnius rezultatus. Žandikaulio ilgio prognozei buvo gauti 0.48 determinacijos koeficientas viršutiniam žandikauliui ir 0.48 – apatiniam, todėl galime teigti, jog modeliai pakankamai tiksliai aprašo turimus duomenis. Sprendimų medžiai, prognozuojantys lytį, parodė, jog viršutinio žandikaulio AUC lygus 0.81, o apatinio – 0.65, kas reiškia, jog viršutinio žandikaulio modelis veikia gerai, bet apatinio žandikaulio modelis veikia silpnai. Taip pat buvo sudarytas atsitiktinis miškas, kuris prognozuoja rudosios lapės lytį. Šie modeliai pateikė geresnius AUC rezultatus, nei sprendimų medžiai. Viršutinio žandikaulio AUC lygus 0.85, o apatinio – 0.86. Tai indikuoja puikų modelių veikimą.

Kadangi tiek tiesinės regresijos modeliai, tiek logistinės regresijos modeliai nėra patikimi prognozuoti žandikaulio ilgį ir lapės lyti multikolinearumo problemą būtų galima bandyti spręsti naudojant pagrindinių komponenčių analize arba atsisakant statistiškai nereikšmingų kintamųjų.

Taigi, remiantis atliktomis analizėmis, galime teigti, jog sprendimų medis yra pranašiausias prognozuojant rudosios lapės žandikaulio ilgį, o atsitiktinis miškis buvo pranašiausias prognozuojant lapės lytį.

Literatūra

[1] Red Fox, National Geographic Animals.
https://www.nationalgeographic.com/animals/mammals/facts/red-fox

[2] E. Jurgelėnas, I. Jasinevičiūtė, S. Kerzienė, L. Daugnora, Morphometric analysis of the red fox molar teeth in Lithuania, *Anatomia, Histologia, Embryologia*, 2022, p.p. 452-458.

[3] Classification: ROC Curve and AUC, Machine Learning Crash Course.
https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

[4] R. Draelos, Measuring Performance: AUC & AUROC, Glass Box, 2019.
https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/

[5] Decision Tree, GeeksforGeeks.
https://www.geeksforgeeks.org/decision-tree/

[6] K. Mali, Everything You Need to Know About Linear Regression, Analytics Vidhya, 2021.
https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/

[7] Logistic Regression, IBM.
https://www.ibm.com/topics/logistic-regression

[8] S. Turney, Coefficient of Determination, Scribbr, 2022.
https://www.scribbr.com/statistics/coefficient-of-determination/

[9] E. R. Sruthi, Understanding Random Forest, Analytics Vidhya, 2021.
https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

[10] Available Diagnostic Visualizations, TIBCO Spotfire.
https://docs.tibco.com/pub/spotfire/6.5.2/doc/html/prd/prd_available_diagnostic_visualizations.htm

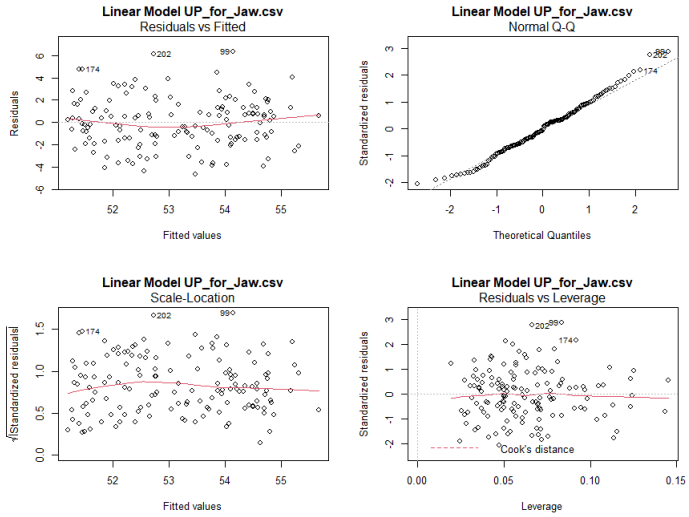
[11] Understanding Q-Q Plots, University of Virginia Library.
https://data.library.virginia.edu/understanding-q-q-plots/

[12] Scale-Location Plot, Statology, 2020.
https://www.statology.org/scale-location-plot/

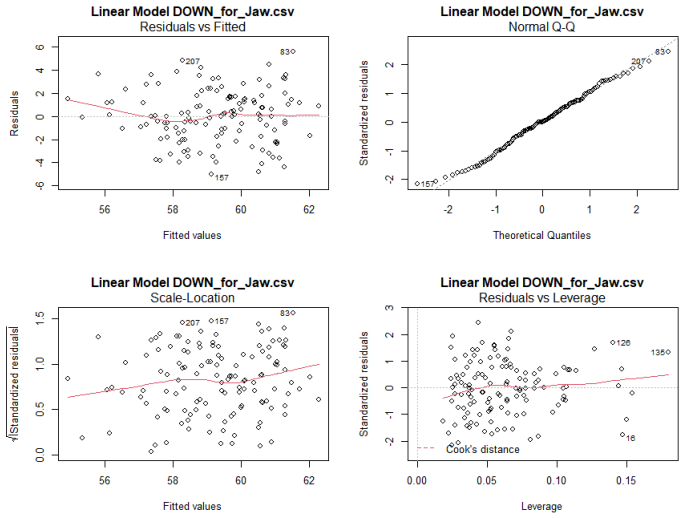
[13] What is a Residuals vs. Leverage Plot? (Definition & Example), Statology, 2020.
https://www.statology.org/residuals-vs-leverage-plot/

[14] G. Martini, Predicted vs. observed values, ResearchGate, 2021.
https://www.researchgate.net/figure/Predicted-vs-observed-values-Each-plot-shows-the-predicted-value-obtained-as-the-median_fig1_352958568

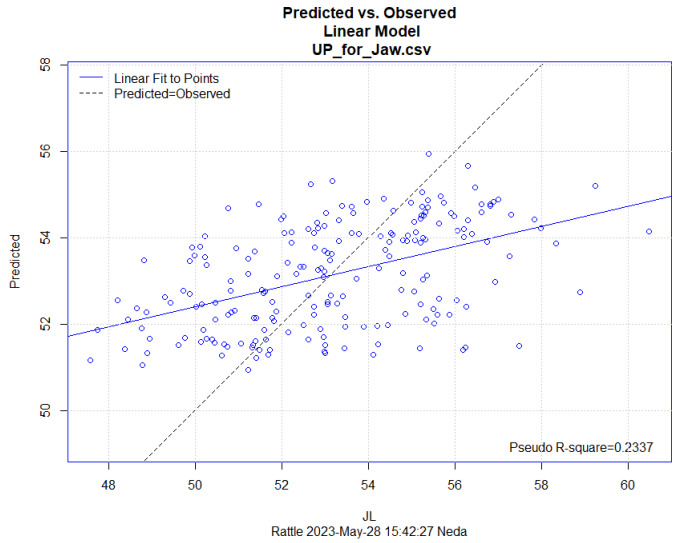
5 A priedas



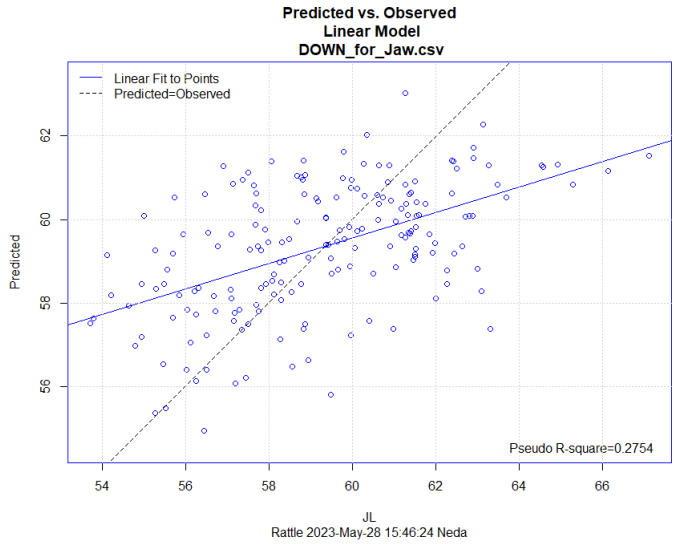
19 pav.: Viršutinio žandikaulio krūminių dantų osteometrijos rezultatų tiesinės regresijos grafikai pradiniams duomenims.



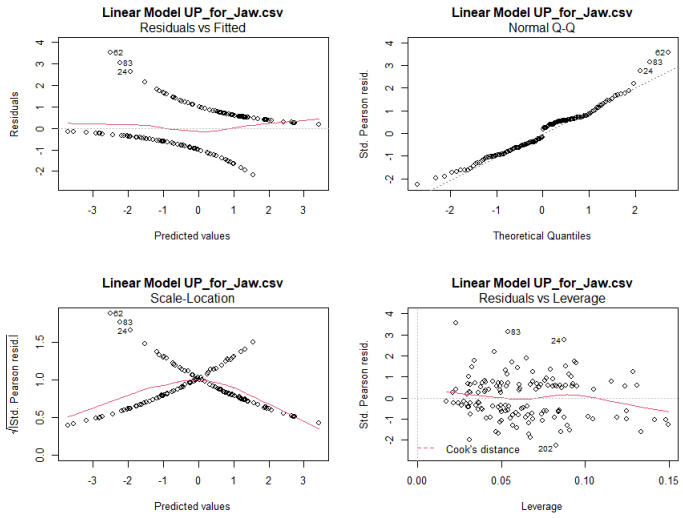
20 pav.: Apatinio žandikaulio krūminių dantų osteometrijos rezultatų tiesinės regresijos grafikai pradiniams duomenims.



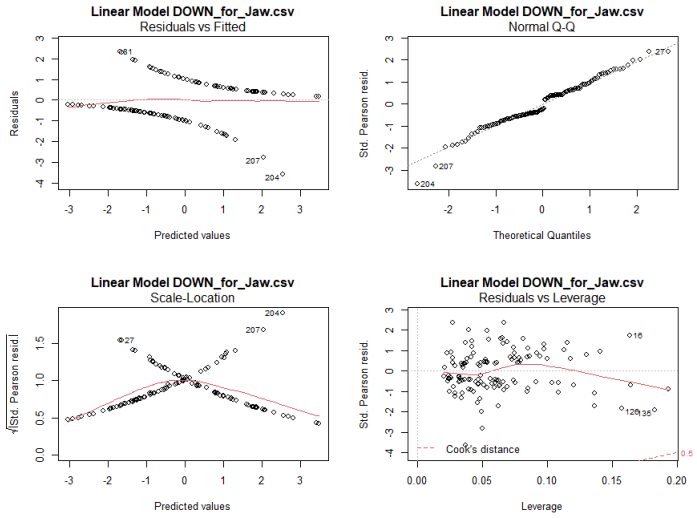
21 pav.: Viršutinio žandikaulio krūminių dantų osteometrijos rezultatų tiesinės regresijos „Predicted vs. observed" grafikas pradiniams duomenims.



22 pav.: Apatinio žandikaulio krūminių dantų osteometrijos rezultatų tiesinės regresijos „Predicted vs. observed" grafikas pradiniams duomenims.



23 pav.: Viršutinio žandikaulio krūminių dantų osteometrijos rezultatų logistinės regresijos grafikai pradiniams duomenims.



24 pav.: Apatinio žandikaulio krūminių dantų osteometrijos rezultatų logistinės regresijos grafikai pradiniams duomenims.

	Numatoma F (proc.)	Numatoma M (proc.)	Klaida (proc.)
Faktinė F (proc.)	36.9	13.3	26.5
Faktinė M (proc.)	13.3	36.5	26.7

5 lentelė: Klaidų matrica viršutinio žandikaulio krūminių dantų osteometrijos rezultatų tiesinei regresijai pradiniams duomenims.

	Numatoma F (proc.)	Numatoma M (proc.)	Klaida (proc.)
Faktinė F (proc.)	38.8	12.6	24.5
Faktinė M (proc.)	14.8	33.9	30.3

6 lentelė: Klaidų matrica apatinio žandikaulio krūminių dantų osteometrijos rezultatų tiesinei regresijai pradiniams duomenims.

6 B priedas

R kodas:

```
#=====================================================================

summary(UP)
summary(DOWN)

#statistic / only need to change column name and dataset(UP/DOWN)

mean ( DOWN$M3_L_to_B_ratio )
median ( DOWN$M3_L_to_B_ratio )
sqrt ( var ( DOWN$M3_L_to_B_ratio ) )
max( DOWN$M3_L_to_B_ratio )
min ( DOWN$M3_L_to_B_ratio )

#how to open Rattle

install.packages("RGtk2")
install.packages("https://cran.microsoft.com/snapshot/2021-12-15/
                 bin/windows/contrib/4.1/RGtk2_2.20.36.2.zip", repos=NULL)
remotes::install_github("cran/rattle")
library(rattle)
rattle()

#=====================================================================

# Rattle is Copyright (c) 2006-2021 Togaware Pty Ltd.
# It is free (as in libre) open source software.
# It is licensed under the GNU General Public License,
# Version 2. Rattle comes with ABSOLUTELY NO WARRANTY.
# Rattle was written by Graham Williams with contributions
# from others as acknowledged in 'library(help=rattle)'.
# Visit https://rattle.togaware.com/ for details.

#=====================================================================

# Rattle version 5.5.1 user 'Neda'
# This log captures interactions with Rattle as an R script.

# For repeatability, export this activity log to a
# file, like 'model.R' using the Export button or
# through the Tools menu. Th script can then serve as a
# starting point for developing your own scripts.
# After xporting to a file called 'model.R', for exmample,
# you can type into a new R Console the command
# "source('model.R')" and so repeat all actions. Generally,
```

```
# you will want to edit the file to suit your own needs.
# You can also edit this log in place to record additional
# information before exporting the script.

# Note that saving/loading projects retains this log.

# We begin most scripts by loading the required packages.
# Here are some initial packages to load and others will be
# identified as we proceed through the script. When writing
# our own scripts we often collect together the library
# commands at the beginning of the script here.

library(rattle)   # Access the weather dataset and utilities.
library(magrittr) # Utilise %>% and %<>% pipeline operators.

# This log generally records the process of building a model.
# However, with very little effort the log can also be used
# to score a new dataset. The logical variable 'building'
# is used to toggle between generating transformations,
# when building a model and using the transformations,
# when scoring a dataset.

building <- TRUE
scoring  <- ! building

# A pre-defined value is used to reset the random seed
# so that results are repeatable.

crv$seed <- 42

#=======================================================================

# Load a dataset from file.

fname         <- "file:///C:/Users/Neda/Desktop/models/UP_for_Jaw.csv"
crs$dataset <- read.csv(fname,
na.strings=c(".", "NA", "", "?"),
strip.white=TRUE, encoding="UTF-8")

#=======================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=230 train=161 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
```

```
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("NR", "P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL", "sex", "P4_area", "M1_area",
                   "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$numeric   <- c("NR", "P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL", "sex", "P4_area", "M1_area",
                   "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$categoric <- NULL
crs$target    <- "Sex_Categorical"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- NULL
crs$weights   <- NULL
#============================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=230 train=161 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.

crs$input     <- c("P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "sex", "P4_area", "M1_area",
                   "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio")
```

```
crs$numeric   <- c("P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "sex", "P4_area", "M1_area",
                   "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$categoric <- NULL
crs$target    <- "JL"
crs$risk      <- NULL
crs$ident     <- "NR"
crs$ignore    <- "Sex_Categorical"
crs$weights   <- NULL
#=========================================================================
# Regression model
# Build a Regression model.
crs$glm <- lm(JL ~ ., data=crs$dataset[crs$train,c(crs$input, crs$target)])
# Generate a textual view of the Linear model.
print(summary(crs$glm))
cat('==== ANOVA ====
')
print(anova(crs$glm))
print("
")
# Plot the model evaluation.
ttl <- genPlotTitleCmd("Linear Model",crs$dataname,vector=TRUE)
plot(crs$glm, main=ttl[1])
#=========================================================================
# Evaluate model performance on the full dataset.
# GLM: Generate a Predicted v Observed plot for glm model on UP_for_Jaw.csv.
crs$pr <- predict(crs$glm,
   type    = "response",
   newdata = crs$dataset[c(crs$input, crs$target)])
# Obtain the observed output for the dataset.
obs <- subset(crs$dataset[c(crs$input, crs$target)], select=crs$target)
# Handle in case categoric target treated as numeric.
obs.rownames <- rownames(obs)
obs <- as.numeric(obs[[1]])
obs <- data.frame(JL=obs)
rownames(obs) <- obs.rownames
# Combine the observed values with the predicted.
fitpoints <- na.omit(cbind(obs, Predicted=crs$pr))
# Obtain the pseudo R2 - a correlation.
fitcorr <- format(cor(fitpoints[,1], fitpoints[,2])^2, digits=4)
# Plot settings for the true points and best fit.
op <- par(c(lty="solid", col="blue"))
# Display the observed (X) versus predicted (Y) points.
plot(fitpoints[[1]], fitpoints[[2]], asp=1, xlab="JL", ylab="Predicted")
# Generate a simple linear fit between predicted and observed.
```

```
prline <- lm(fitpoints[,2] ~ fitpoints[,1])
# Add the linear fit to the plot.
abline(prline)
# Add a diagonal representing perfect correlation.
par(c(lty="dashed", col="black"))
abline(0, 1)
# Include a pseudo R-square on the plot
legend("bottomright",  sprintf(" Pseudo R-square=%s ", fitcorr),  bty="n")
# Add a title and grid to the plot.
title(main="Predicted vs. Observed
 Linear Model
 UP_for_Jaw.csv",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["user"]))
grid()
#=======================================================================
# Decision Tree
# The 'rpart' package provides the 'rpart' function.
library(rpart, quietly=TRUE)
# Reset the random number seed to obtain the same results each time.
set.seed(crv$seed)
# Build the Decision Tree model.
crs$rpart <- rpart(JL ~ .,
    data=crs$dataset[crs$train, c(crs$input, crs$target)],
    method="anova",
    parms=list(split="information"),
      control=rpart.control(minbucket=4,
        usesurrogate=0,
        maxsurrogate=0),
    model=TRUE)
# Generate a textual view of the Decision Tree model.
print(crs$rpart)
printcp(crs$rpart)
cat("\n")
#=======================================================================
# Plot the resulting Decision Tree.
# We use the rpart.plot package.
fancyRpartPlot(crs$rpart, main="Decision Tree UP_for_Jaw.csv $ JL")
#=======================================================================
# Evaluate model performance on the full dataset.
# RPART: Generate a Predicted v Observed plot for rpart model on UP_for_Jaw.csv.
crs$pr <- predict(crs$rpart, newdata=crs$dataset[c(crs$input, crs$target)])
# Obtain the observed output for the dataset.
obs <- subset(crs$dataset[c(crs$input, crs$target)], select=crs$target)
# Handle in case categoric target treated as numeric.
obs.rownames <- rownames(obs)
obs <- as.numeric(obs[[1]])
obs <- data.frame(JL=obs)
```

```
rownames(obs) <- obs.rownames
# Combine the observed values with the predicted.
fitpoints <- na.omit(cbind(obs, Predicted=crs$pr))
# Obtain the pseudo R2 - a correlation.
fitcorr <- format(cor(fitpoints[,1], fitpoints[,2])^2, digits=4)
# Plot settings for the true points and best fit.
op <- par(c(lty="solid", col="blue"))
# Display the observed (X) versus predicted (Y) points.
plot(fitpoints[[1]], fitpoints[[2]], asp=1, xlab="JL", ylab="Predicted")
# Generate a simple linear fit between predicted and observed.
prline <- lm(fitpoints[,2] ~ fitpoints[,1])
# Add the linear fit to the plot.
abline(prline)
# Add a diagonal representing perfect correlation.
par(c(lty="dashed", col="black"))
abline(0, 1)
# Include a pseudo R-square on the plot
legend("bottomright",  sprintf(" Pseudo R-square=%s ", fitcorr),  bty="n")
# Add a title and grid to the plot.
title(main="Predicted vs. Observed
 Decision Tree Model
 UP_for_Jaw.csv",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["user"]))
grid()
#=======================================================================
library(rattle)   # Access the weather dataset and utilities.
library(magrittr) # Utilise %>% and %<>% pipeline operators.
building <- TRUE
scoring  <- ! building
crv$seed <- 42
#=======================================================================
# Load a dataset from file.
fname         <- "file:///C:/Users/Neda/Desktop/models/DOWN_for_Jaw.csv"
crs$dataset <- read.csv(fname,
na.strings=c(".", "NA", "", "?"),
strip.white=TRUE, encoding="UTF-8")
#=======================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=229 train=160 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
```

```
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("NR", "M1_L", "M1_B", "M2_L", "M2_B", "M3_L",
                   "M3_B", "JL", "sex", "M1_area", "M2_area",
                   "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")
crs$numeric   <- c("NR", "M1_L", "M1_B", "M2_L", "M2_B", "M3_L",
                   "M3_B", "JL", "sex", "M1_area", "M2_area",
                   "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")
crs$categoric <- NULL
crs$target    <- "sex_categorical"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- NULL
crs$weights   <- NULL
#============================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=229 train=160 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
                   "sex", "M1_area", "M2_area", "M3_area",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")
crs$numeric   <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
                   "sex", "M1_area", "M2_area", "M3_area",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")
```

```
crs$categoric <- NULL
crs$target    <- "JL"
crs$risk      <- NULL
crs$ident     <- "NR"
crs$ignore    <- "sex_categorical"
crs$weights   <- NULL
#============================================================
# Regression model
# Build a Regression model.
crs$glm <- lm(JL ~ ., data=crs$dataset[crs$train,c(crs$input, crs$target)])
# Generate a textual view of the Linear model.
print(summary(crs$glm))
cat('==== ANOVA ====
')
print(anova(crs$glm))
print("
")
# Plot the model evaluation.
ttl <- genPlotTitleCmd("Linear Model",crs$dataname,vector=TRUE)
plot(crs$glm, main=ttl[1])
#============================================================
# Evaluate model performance on the full dataset.
# GLM: Generate a Predicted v Observed plot for glm model on DOWN_for_Jaw.csv.
crs$pr <- predict(crs$glm,
   type    = "response",
   newdata = crs$dataset[c(crs$input, crs$target)])
# Obtain the observed output for the dataset.
obs <- subset(crs$dataset[c(crs$input, crs$target)], select=crs$target)
# Handle in case categoric target treated as numeric.
obs.rownames <- rownames(obs)
obs <- as.numeric(obs[[1]])
obs <- data.frame(JL=obs)
rownames(obs) <- obs.rownames
# Combine the observed values with the predicted.
fitpoints <- na.omit(cbind(obs, Predicted=crs$pr))
# Obtain the pseudo R2 - a correlation.
fitcorr <- format(cor(fitpoints[,1], fitpoints[,2])^2, digits=4)
# Plot settings for the true points and best fit.
op <- par(c(lty="solid", col="blue"))
# Display the observed (X) versus predicted (Y) points.
plot(fitpoints[[1]], fitpoints[[2]], asp=1, xlab="JL", ylab="Predicted")
# Generate a simple linear fit between predicted and observed.
prline <- lm(fitpoints[,2] ~ fitpoints[,1])
# Add the linear fit to the plot.
abline(prline)
# Add a diagonal representing perfect correlation.
par(c(lty="dashed", col="black"))
```

```
abline(0, 1)
# Include a pseudo R-square on the plot
legend("bottomright",  sprintf(" Pseudo R-square=%s ", fitcorr),  bty="n")
# Add a title and grid to the plot.
title(main="Predicted vs. Observed
 Linear Model
 DOWN_for_Jaw.csv",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["user"]))
grid()
#=======================================================================
# Decision Tree
# The 'rpart' package provides the 'rpart' function.
library(rpart, quietly=TRUE)
# Reset the random number seed to obtain the same results each time.
set.seed(crv$seed)
# Build the Decision Tree model.
crs$rpart <- rpart(JL ~ .,
    data=crs$dataset[crs$train, c(crs$input, crs$target)],
    method="anova",
    parms=list(split="information"),
      control=rpart.control(minbucket=3,
        usesurrogate=0,
        maxsurrogate=0),
    model=TRUE)
# Generate a textual view of the Decision Tree model.
print(crs$rpart)
printcp(crs$rpart)
cat("\n")
#=======================================================================
# Plot the resulting Decision Tree.
# We use the rpart.plot package.
fancyRpartPlot(crs$rpart, main="Decision Tree DOWN_for_Jaw.csv $ JL")
#=======================================================================
# Evaluate model performance on the full dataset.
# RPART: Generate a Predicted v Observed plot for rpart model on DOWN_for_Jaw.csv.
crs$pr <- predict(crs$rpart, newdata=crs$dataset[c(crs$input, crs$target)])
# Obtain the observed output for the dataset.
obs <- subset(crs$dataset[c(crs$input, crs$target)], select=crs$target)
# Handle in case categoric target treated as numeric.
obs.rownames <- rownames(obs)
obs <- as.numeric(obs[[1]])
obs <- data.frame(JL=obs)
rownames(obs) <- obs.rownames
# Combine the observed values with the predicted.
fitpoints <- na.omit(cbind(obs, Predicted=crs$pr))
# Obtain the pseudo R2 - a correlation.
fitcorr <- format(cor(fitpoints[,1], fitpoints[,2])^2, digits=4)
```

```
# Plot settings for the true points and best fit.
op <- par(c(lty="solid", col="blue"))
# Display the observed (X) versus predicted (Y) points.
plot(fitpoints[[1]], fitpoints[[2]], asp=1, xlab="JL", ylab="Predicted")
# Generate a simple linear fit between predicted and observed.
prline <- lm(fitpoints[,2] ~ fitpoints[,1])
# Add the linear fit to the plot.
abline(prline)
# Add a diagonal representing perfect correlation.
par(c(lty="dashed", col="black"))
abline(0, 1)
# Include a pseudo R-square on the plot
legend("bottomright",  sprintf(" Pseudo R-square=%s ", fitcorr),  bty="n")
# Add a title and grid to the plot.
title(main="Predicted vs. Observed
 Decision Tree Model
 DOWN_for_Jaw.csv",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["user"]))
grid()
# GLM: Generate a Predicted v Observed plot for glm model on DOWN_for_Jaw.csv.
crs$pr <- predict(crs$glm,
   type    = "response",
   newdata = crs$dataset[c(crs$input, crs$target)])
# Obtain the observed output for the dataset.
obs <- subset(crs$dataset[c(crs$input, crs$target)], select=crs$target)
# Handle in case categoric target treated as numeric.
obs.rownames <- rownames(obs)
obs <- as.numeric(obs[[1]])
obs <- data.frame(JL=obs)
rownames(obs) <- obs.rownames
# Combine the observed values with the predicted.
fitpoints <- na.omit(cbind(obs, Predicted=crs$pr))
# Obtain the pseudo R2 - a correlation.
fitcorr <- format(cor(fitpoints[,1], fitpoints[,2])^2, digits=4)
# Plot settings for the true points and best fit.
op <- par(c(lty="solid", col="blue"))
# Display the observed (X) versus predicted (Y) points.
plot(fitpoints[[1]], fitpoints[[2]], asp=1, xlab="JL", ylab="Predicted")
# Generate a simple linear fit between predicted and observed.
prline <- lm(fitpoints[,2] ~ fitpoints[,1])
# Add the linear fit to the plot.
abline(prline)
# Add a diagonal representing perfect correlation.
par(c(lty="dashed", col="black"))
abline(0, 1)
# Include a pseudo R-square on the plot
legend("bottomright",  sprintf(" Pseudo R-square=%s ", fitcorr),  bty="n")
```

```
# Add a title and grid to the plot.
title(main="Predicted vs. Observed
 Linear Model
 DOWN_for_Jaw.csv",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["user"]))
grid()
#=======================================================================
# Evaluate model performance on the full dataset.
# RPART: Generate a Predicted v Observed plot for rpart model on DOWN_for_Jaw.csv.
crs$pr <- predict(crs$rpart, newdata=crs$dataset[c(crs$input, crs$target)])
# Obtain the observed output for the dataset.
obs <- subset(crs$dataset[c(crs$input, crs$target)], select=crs$target)
# Handle in case categoric target treated as numeric.
obs.rownames <- rownames(obs)
obs <- as.numeric(obs[[1]])
obs <- data.frame(JL=obs)
rownames(obs) <- obs.rownames
# Combine the observed values with the predicted.
fitpoints <- na.omit(cbind(obs, Predicted=crs$pr))
# Obtain the pseudo R2 - a correlation.
fitcorr <- format(cor(fitpoints[,1], fitpoints[,2])^2, digits=4)
# Plot settings for the true points and best fit.
op <- par(c(lty="solid", col="blue"))
# Display the observed (X) versus predicted (Y) points.
plot(fitpoints[[1]], fitpoints[[2]], asp=1, xlab="JL", ylab="Predicted")
# Generate a simple linear fit between predicted and observed.
prline <- lm(fitpoints[,2] ~ fitpoints[,1])
# Add the linear fit to the plot.
abline(prline)
# Add a diagonal representing perfect correlation.
par(c(lty="dashed", col="black"))
abline(0, 1)
# Include a pseudo R-square on the plot
legend("bottomright",  sprintf(" Pseudo R-square=%s ", fitcorr),  bty="n")
# Add a title and grid to the plot.
title(main="Predicted vs. Observed
 Decision Tree Model
 DOWN_for_Jaw.csv",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["user"]))
grid()
#=======================================================================
library(rattle)   # Access the weather dataset and utilities.
library(magrittr) # Utilise %>% and %<>% pipeline operators.
building <- TRUE
scoring  <- ! building
crv$seed <- 42
```

```
#=======================================================================
# Load a dataset from file.
fname         <- "file:///C:/Users/Neda/Desktop/models/UP_for_Jaw.csv"
crs$dataset <- read.csv(fname,
na.strings=c(".", "NA", "", "?"),
strip.white=TRUE, encoding="UTF-8")
#=======================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=230 train=161 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("NR", "P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL", "sex", "P4_area", "M1_area",
                   "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$numeric   <- c("NR", "P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL", "sex", "P4_area", "M1_area",
                   "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$categoric <- NULL
crs$target    <- "Sex_Categorical"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- NULL
crs$weights   <- NULL
#=======================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=230 train=161 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
```

```
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL", "P4_area", "M1_area",
                   "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$numeric   <- c("P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL", "P4_area", "M1_area",
                   "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$categoric <- NULL
crs$target    <- "Sex_Categorical"
crs$risk      <- NULL
crs$ident     <- "NR"
crs$ignore    <- "sex"
crs$weights   <- NULL
#============================================================================
# Regression model
# Build a Regression model.
crs$glm <- glm(Sex_Categorical ~ .,
    data=crs$dataset[crs$train, c(crs$input, crs$target)],
    family=binomial(link="logit"))
# Generate a textual view of the Linear model.
print(summary(crs$glm))
cat(sprintf("Log likelihood: %.3f (%d df)\n",
            logLik(crs$glm)[1],
            attr(logLik(crs$glm), "df")))
cat(sprintf("Null/Residual deviance difference: %.3f (%d df)\n",
            crs$glm$null.deviance-crs$glm$deviance,
            crs$glm$df.null-crs$glm$df.residual))
cat(sprintf("Chi-square p-value: %.8f\n",
            dchisq(crs$glm$null.deviance-crs$glm$deviance,
                   crs$glm$df.null-crs$glm$df.residual)))
cat(sprintf("Pseudo R-Square (optimistic): %.8f\n",
             cor(crs$glm$y, crs$glm$fitted.values)))
cat('\n==== ANOVA ====\n\n')
print(anova(crs$glm, test="Chisq"))
cat("\n")
# Plot the model evaluation.
ttl <- genPlotTitleCmd("Linear Model",crs$dataname,vector=TRUE)
```

```
plot(crs$glm, main=ttl[1])
#=====================================================================
# Evaluate model performance on the full dataset.
# Generate an Error Matrix for the Linear model.
# Obtain the response from the Linear model.
crs$pr <- as.vector(ifelse(predict(crs$glm,
   type    = "response",
   newdata = crs$dataset[c(crs$input, crs$target)]) > 0.5, "M", "F"))
# Generate the confusion matrix showing counts.
rattle::errorMatrix(crs$dataset[c(crs$input, crs$target)]$Sex_Categorical, crs$pr, count=TRUE)
# Generate the confusion matrix showing proportions.
(per <- rattle::errorMatrix(crs$dataset[c(crs$input, crs$target)]$Sex_Categorical, crs$pr))
# Calculate the overall error percentage.
cat(100-sum(diag(per), na.rm=TRUE))
# Calculate the averaged class error percentage.
cat(mean(per[,"Error"], na.rm=TRUE))
#=====================================================================
# Decision Tree
# The 'rpart' package provides the 'rpart' function.
library(rpart, quietly=TRUE)
# Reset the random number seed to obtain the same results each time.
set.seed(crv$seed)
# Build the Decision Tree model.
crs$rpart <- rpart(Sex_Categorical ~ .,
    data=crs$dataset[crs$train, c(crs$input, crs$target)],
    method="class",
    parms=list(split="information"),
      control=rpart.control(minbucket=3,
        usesurrogate=0,
        maxsurrogate=0),
    model=TRUE)
# Generate a textual view of the Decision Tree model.
print(crs$rpart)
printcp(crs$rpart)
cat("\n")
#=====================================================================
# Evaluate model performance on the full dataset.
# ROC Curve: requires the ROCR package.
library(ROCR)
# ROC Curve: requires the ggplot2 package.
library(ggplot2, quietly=TRUE)
# Generate an ROC Curve for the rpart model on UP_for_Jaw.csv.
crs$pr <- predict(crs$rpart, newdata=crs$dataset[c(crs$input, crs$target)])[,2]
# Remove observations with missing target.
no.miss   <- na.omit(crs$dataset[c(crs$input, crs$target)]$Sex_Categorical)
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL
```

```
if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}
pe <- performance(pred, "tpr", "fpr")
au <- performance(pred, "auc")@y.values[[1]]
pd <- data.frame(fpr=unlist(pe@x.values), tpr=unlist(pe@y.values))
p <- ggplot(pd, aes(x=fpr, y=tpr))
p <- p + geom_line(colour="red")
p <- p + xlab("False Positive Rate") + ylab("True Positive Rate")
p <- p + ggtitle("ROC Curve Decision Tree UP_for_Jaw.csv Sex_Categorical")
p <- p + theme(plot.title=element_text(size=10))
p <- p + geom_line(data=data.frame(), aes(x=c(0,1), y=c(0,1)), colour="grey")
p <- p + annotate("text", x=0.50, y=0.00, hjust=0, vjust=0, size=5,
                  label=paste("AUC =", round(au, 2)))
print(p)
# Calculate the area under the curve for the plot.
# Remove observations with missing target.
no.miss   <- na.omit(crs$dataset[c(crs$input, crs$target)]$Sex_Categorical)
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL
if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}
performance(pred, "auc")
#=====================================================================
# Plot the resulting Decision Tree.
# We use the rpart.plot package.
fancyRpartPlot(crs$rpart, main="Decision Tree UP_for_Jaw.csv $ Sex_Categorical")
#=====================================================================
library(rattle)   # Access the weather dataset and utilities.
library(magrittr) # Utilise %>% and %<>% pipeline operators.
building <- TRUE
scoring  <- ! building
crv$seed <- 42
#=====================================================================
# Load a dataset from file.
fname         <- "file:///C:/Users/Neda/Desktop/models/DOWN_for_Jaw.csv"
crs$dataset <- read.csv(fname,
na.strings=c(".", "NA", "", "?"),
```

```
strip.white=TRUE, encoding="UTF-8")
#=====================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=229 train=160 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("NR", "M1_L", "M1_B", "M2_L", "M2_B", "M3_L",
                   "M3_B", "JL", "sex", "M1_area", "M2_area",
                   "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")
crs$numeric   <- c("NR", "M1_L", "M1_B", "M2_L", "M2_B", "M3_L",
                   "M3_B", "JL", "sex", "M1_area", "M2_area",
                   "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")
crs$categoric <- NULL
crs$target    <- "sex_categorical"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- NULL
crs$weights   <- NULL
#=====================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=229 train=160 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
```

```
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
                   "JL", "M1_area", "M2_area", "M3_area",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")
crs$numeric   <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
                   "JL", "M1_area", "M2_area", "M3_area",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")
crs$categoric <- NULL
crs$target    <- "sex_categorical"
crs$risk      <- NULL
crs$ident     <- "NR"
crs$ignore    <- "sex"
crs$weights   <- NULL
#============================================================================
# Regression model
# Build a Regression model.
crs$glm <- glm(sex_categorical ~ .,
    data=crs$dataset[crs$train, c(crs$input, crs$target)],
    family=binomial(link="logit"))
# Generate a textual view of the Linear model.
print(summary(crs$glm))
cat(sprintf("Log likelihood: %.3f (%d df)\n",
            logLik(crs$glm)[1],
            attr(logLik(crs$glm), "df")))
cat(sprintf("Null/Residual deviance difference: %.3f (%d df)\n",
            crs$glm$null.deviance-crs$glm$deviance,
            crs$glm$df.null-crs$glm$df.residual))
cat(sprintf("Chi-square p-value: %.8f\n",
            dchisq(crs$glm$null.deviance-crs$glm$deviance,
                   crs$glm$df.null-crs$glm$df.residual)))
cat(sprintf("Pseudo R-Square (optimistic): %.8f\n",
             cor(crs$glm$y, crs$glm$fitted.values)))
cat('\n==== ANOVA ====\n\n')
print(anova(crs$glm, test="Chisq"))
cat("\n")
# Plot the model evaluation.
ttl <- genPlotTitleCmd("Linear Model",crs$dataname,vector=TRUE)
plot(crs$glm, main=ttl[1])
#============================================================================
# Evaluate model performance on the full dataset.
# Generate an Error Matrix for the Linear model.
# Obtain the response from the Linear model.
```

```
crs$pr <- as.vector(ifelse(predict(crs$glm,
   type    = "response",
   newdata = crs$dataset[c(crs$input, crs$target)]) > 0.5, "M", "F"))
# Generate the confusion matrix showing counts.
rattle::errorMatrix(crs$dataset[c(crs$input, crs$target)]$sex_categorical, crs$pr, count=TRUE)
# Generate the confusion matrix showing proportions.
(per <- rattle::errorMatrix(crs$dataset[c(crs$input, crs$target)]$sex_categorical, crs$pr))
# Calculate the overall error percentage.
cat(100-sum(diag(per), na.rm=TRUE))
# Calculate the averaged class error percentage.
cat(mean(per[,"Error"], na.rm=TRUE))
#=====================================================================
# Decision Tree
# The 'rpart' package provides the 'rpart' function.
library(rpart, quietly=TRUE)
# Reset the random number seed to obtain the same results each time.
set.seed(crv$seed)
# Build the Decision Tree model.
crs$rpart <- rpart(sex_categorical ~ .,
    data=crs$dataset[crs$train, c(crs$input, crs$target)],
    method="class",
    parms=list(split="information"),
      control=rpart.control(minbucket=3,
        usesurrogate=0,
        maxsurrogate=0),
    model=TRUE)
# Generate a textual view of the Decision Tree model.
print(crs$rpart)
printcp(crs$rpart)
cat("\n")
#=====================================================================
# Plot the resulting Decision Tree.
# We use the rpart.plot package.
fancyRpartPlot(crs$rpart, main="Decision Tree DOWN_for_Jaw.csv $ sex_categorical")
#=====================================================================
# Evaluate model performance on the full dataset.
# ROC Curve: requires the ROCR package.
library(ROCR)
# ROC Curve: requires the ggplot2 package.
library(ggplot2, quietly=TRUE)
# Generate an ROC Curve for the rpart model on DOWN_for_Jaw.csv.
crs$pr <- predict(crs$rpart, newdata=crs$dataset[c(crs$input, crs$target)])[,2]
# Remove observations with missing target.
no.miss   <- na.omit(crs$dataset[c(crs$input, crs$target)]$sex_categorical)
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL
if (length(miss.list))
```

```
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}
pe <- performance(pred, "tpr", "fpr")
au <- performance(pred, "auc")@y.values[[1]]
pd <- data.frame(fpr=unlist(pe@x.values), tpr=unlist(pe@y.values))
p <- ggplot(pd, aes(x=fpr, y=tpr))
p <- p + geom_line(colour="red")
p <- p + xlab("False Positive Rate") + ylab("True Positive Rate")
p <- p + ggtitle("ROC Curve Decision Tree DOWN_for_Jaw.csv sex_categorical")
p <- p + theme(plot.title=element_text(size=10))
p <- p + geom_line(data=data.frame(), aes(x=c(0,1), y=c(0,1)), colour="grey")
p <- p + annotate("text", x=0.50, y=0.00, hjust=0, vjust=0, size=5,
                  label=paste("AUC =", round(au, 2)))
print(p)
# Calculate the area under the curve for the plot.
# Remove observations with missing target.
no.miss   <- na.omit(crs$dataset[c(crs$input, crs$target)]$sex_categorical)
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL
if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}
performance(pred, "auc")
#=============================================================================
# Load a dataset from file.

fname         <- "file:///C:/Users/Neda/Desktop/models/UP_for_Jaw.csv"
crs$dataset <- read.csv(fname,
na.strings=c(".", "NA", "", "?"),
strip.white=TRUE, encoding="UTF-8")
#=============================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=230 train=161 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
```

```
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("NR", "P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL", "sex", "P4_area", "M1_area",
                   "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio")

crs$numeric   <- c("NR", "P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL", "sex", "P4_area", "M1_area",
                   "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$categoric <- NULL
crs$target    <- "Sex_Categorical"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- NULL
crs$weights   <- NULL
#============================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=230 train=161 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL", "P4_area", "M1_area",
                   "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$numeric   <- c("P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
```

```
                  "M2_L", "M2_B", "JL", "P4_area", "M1_area",
                  "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                  "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$categoric <- NULL
crs$target    <- "Sex_Categorical"
crs$risk      <- NULL
crs$ident     <- "NR"
crs$ignore    <- "sex"
crs$weights   <- NULL
#============================================================================
# Build a Random Forest model using the traditional approach.
set.seed(crv$seed)
crs$rf <- randomForest::randomForest(Sex_Categorical ~ .,
  data=crs$dataset[crs$train, c(crs$input, crs$target)],
  ntree=500,
  mtry=15,
  importance=TRUE,
  na.action=randomForest::na.roughfix,
  replace=FALSE)
# Generate textual output of the 'Random Forest' model.
crs$rf
# The 'pROC' package implements various AUC functions.
# Calculate the Area Under the Curve (AUC).
pROC::roc(crs$rf$y, as.numeric(crs$rf$predicted))
# Calculate the AUC Confidence Interval.
pROC::ci.auc(crs$rf$y, as.numeric(crs$rf$predicted))
# List the importance of the variables.
rn <- round(randomForest::importance(crs$rf), 2)
rn[order(rn[,3], decreasing=TRUE),]
#============================================================================
# Evaluate model performance on the full dataset.
# ROC Curve: requires the ROCR package.
library(ROCR)
# ROC Curve: requires the ggplot2 package.
library(ggplot2, quietly=TRUE)
# Generate an ROC Curve for the rf model on UP_for_Jaw.csv.
crs$pr <- predict(crs$rf, newdata=na.omit(crs$dataset[c(crs$input, crs$target)]),
    type    = "prob")[,2]
# Remove observations with missing target.
no.miss   <- na.omit(na.omit(crs$dataset[c(crs$input, crs$target)])$Sex_Categorical)
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL
if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
```

```
  pred <- prediction(crs$pr, no.miss)
}
pe <- performance(pred, "tpr", "fpr")
au <- performance(pred, "auc")@y.values[[1]]
pd <- data.frame(fpr=unlist(pe@x.values), tpr=unlist(pe@y.values))
p <- ggplot(pd, aes(x=fpr, y=tpr))
p <- p + geom_line(colour="red")
p <- p + xlab("False Positive Rate") + ylab("True Positive Rate")
p <- p + ggtitle("ROC Curve Random Forest UP_for_Jaw.csv Sex_Categorical")
p <- p + theme(plot.title=element_text(size=10))
p <- p + geom_line(data=data.frame(), aes(x=c(0,1), y=c(0,1)), colour="grey")
p <- p + annotate("text", x=0.50, y=0.00, hjust=0, vjust=0, size=5,
                  label=paste("AUC =", round(au, 2)))
print(p)
# Calculate the area under the curve for the plot.
# Remove observations with missing target.
no.miss   <- na.omit(na.omit(crs$dataset[c(crs$input, crs$target)])$Sex_Categorical)
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL
if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}
performance(pred, "auc")
#=====================================================================
# Load a dataset from file.
fname         <- "file:///C:/Users/Neda/Desktop/models/DOWN_for_Jaw.csv"
crs$dataset <- read.csv(fname,
na.strings=c(".", "NA", "", "?"),
strip.white=TRUE, encoding="UTF-8")
#=====================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=229 train=160 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
```

```
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("NR", "M1_L", "M1_B", "M2_L", "M2_B", "M3_L",
                   "M3_B", "JL", "sex", "M1_area", "M2_area",
                   "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")
crs$numeric   <- c("NR", "M1_L", "M1_B", "M2_L", "M2_B", "M3_L",
                   "M3_B", "JL", "sex", "M1_area", "M2_area",
                   "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")
crs$categoric <- NULL
crs$target    <- "sex_categorical"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- NULL
crs$weights   <- NULL
#============================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=229 train=160 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
                   "JL", "M1_area", "M2_area", "M3_area",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")

crs$numeric   <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
                   "JL", "M1_area", "M2_area", "M3_area",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")

crs$categoric <- NULL
```

```
crs$target    <- "sex_categorical"
crs$risk      <- NULL
crs$ident     <- "NR"
crs$ignore    <- "sex"
crs$weights   <- NULL
#=====================================================================
# Build a Random Forest model using the traditional approach.
set.seed(crv$seed)
crs$rf <- randomForest::randomForest(sex_categorical ~ .,
  data=crs$dataset[crs$train, c(crs$input, crs$target)],
  ntree=500,
  mtry=13,
  importance=TRUE,
  na.action=randomForest::na.roughfix,
  replace=FALSE)
# Generate textual output of the 'Random Forest' model.
crs$rf
# The 'pROC' package implements various AUC functions.
# Calculate the Area Under the Curve (AUC).
pROC::roc(crs$rf$y, as.numeric(crs$rf$predicted))
# Calculate the AUC Confidence Interval.
pROC::ci.auc(crs$rf$y, as.numeric(crs$rf$predicted))
# List the importance of the variables.
rn <- round(randomForest::importance(crs$rf), 2)
rn[order(rn[,3], decreasing=TRUE),]
#=====================================================================
# Evaluate model performance on the full dataset.
# ROC Curve: requires the ROCR package.
library(ROCR)
# ROC Curve: requires the ggplot2 package.
library(ggplot2, quietly=TRUE)
# Generate an ROC Curve for the rf model on DOWN_for_Jaw.csv.
crs$pr <- predict(crs$rf, newdata=na.omit(crs$dataset[c(crs$input, crs$target)]),
    type    = "prob")[,2]
# Remove observations with missing target.
no.miss   <- na.omit(na.omit(crs$dataset[c(crs$input, crs$target)])$sex_categorical)
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}
pe <- performance(pred, "tpr", "fpr")
```

```
au <- performance(pred, "auc")@y.values[[1]]
pd <- data.frame(fpr=unlist(pe@x.values), tpr=unlist(pe@y.values))
p <- ggplot(pd, aes(x=fpr, y=tpr))
p <- p + geom_line(colour="red")
p <- p + xlab("False Positive Rate") + ylab("True Positive Rate")
p <- p + ggtitle("ROC Curve Random Forest DOWN_for_Jaw.csv sex_categorical")
p <- p + theme(plot.title=element_text(size=10))
p <- p + geom_line(data=data.frame(), aes(x=c(0,1), y=c(0,1)), colour="grey")
p <- p + annotate("text", x=0.50, y=0.00, hjust=0, vjust=0, size=5,
                  label=paste("AUC =", round(au, 2)))
print(p)

# Calculate the area under the curve for the plot.
# Remove observations with missing target.
no.miss   <- na.omit(na.omit(crs$dataset[c(crs$input, crs$target)])$sex_categorical)
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL
if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}
performance(pred, "auc")
########################################
model1 <- lm(JL ~ P4_L+ P4_B+ P4_GB+ M1_L+ M1_B+ M2_L+ M2_B+
               sex+ P4_area+ M1_area+ M2_area+
               P4_L_to_B_ratio+ P4_L_to_GB_ratio+  M1_L_to_B_ratio+
               M2_L_to_B_ratio, data = UP)
vif_values1 <- vif(model1)
print(vif_values1)

model2 <- lm(JL ~ M1_L+ M1_B+ M2_L+ M2_B+ M3_L+ M3_B+
            sex+ M1_area+ M2_area+ M3_area+
            M1_L_to_B_ratio+ M2_L_to_B_ratio+
            M3_L_to_B_ratio, data = DOWN)
vif_values2 <- vif(model2)
print(vif_values2)
########################################
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
```

```
  crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
  crs$test
# The following variable selections have been noted.
crs$input <- c("NR", "P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
               "M2_L", "M2_B", "JL", "sex", "P4_area", "M1_area",
               "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",

               "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$numeric <- c("NR", "P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                 "M2_L", "M2_B", "JL", "sex", "P4_area", "M1_area",
                 "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                 "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$categoric <- NULL
crs$target <- "Sex_Categorical"
crs$risk <- NULL
crs$ident <- NULL
crs$ignore <- NULL
crs$weights <- NULL
#============================================================================
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
  crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
  crs$test
crs$input <- c("P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
               "M2_L", "M2_B", "JL", "P4_area", "M1_area",
               "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
               "M1_L_to_B_ratio", "M2_L_to_B_ratio")

crs$numeric <- c("P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                 "M2_L", "M2_B", "JL", "P4_area", "M1_area",
                 "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                 "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$categoric <- NULL
crs$target <- "Sex_Categorical"
```

```
crs$risk <- NULL
crs$ident <- "NR"
crs$ignore <- "sex"
crs$weights <- NULL
crs$glmUP <- glm(Sex_Categorical ~ .,
                 data=crs$dataset[crs$train, c(crs$input, crs$target)],
                 family=binomial(link="logit"))
vif(crs$glmUP)
#=====================================================================
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
  crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
  crs$test
crs$input <- c("NR", "M1_L", "M1_B", "M2_L", "M2_B", "M3_L",
               "M3_B", "JL", "sex", "M1_area", "M2_area",
               "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio",
               "M3_L_to_B_ratio")
crs$numeric <- c("NR", "M1_L", "M1_B", "M2_L", "M2_B", "M3_L",
                 "M3_B", "JL", "sex", "M1_area", "M2_area",
                 "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                 "M3_L_to_B_ratio")
crs$categoric <- NULL
crs$target <- "sex_categorical"
crs$risk <- NULL
crs$ident <- NULL
crs$ignore <- NULL
crs$weights <- NULL
#=====================================================================
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
  crs$validate
crs$nobs %>%
  seq_len() %>%
```

```
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
  crs$test
# The following variable selections have been noted.
crs$input <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
               "JL", "M1_area", "M2_area", "M3_area",
               "M1_L_to_B_ratio", "M2_L_to_B_ratio",
               "M3_L_to_B_ratio")
crs$numeric <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
                 "JL", "M1_area", "M2_area", "M3_area",
                 "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                 "M3_L_to_B_ratio")
crs$categoric <- NULL
crs$target <- "sex_categorical"
crs$risk <- NULL
crs$ident <- "NR"
crs$ignore <- "sex"
crs$weights <- NULL
#============================================================================
crs$glmDOWN <- glm(sex_categorical ~ .,
               data=crs$dataset[crs$train, c(crs$input, crs$target)],
               family=binomial(link="logit"))

vif(crs$glmDOWN)
building <- TRUE
scoring  <- ! building
# A pre-defined value is used to reset the random seed
# so that results are repeatable.
crv$seed <- 42
#============================================================================
# Load a dataset from file.

fname         <- "file:///C:/Users/Neda/Desktop/models/UP_for_Jaw.csv"
crs$dataset <- read.csv(fname,
na.strings=c(".", "NA", "", "?"),
strip.white=TRUE, encoding="UTF-8")
#============================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=230 train=161 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
```

```
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("NR", "P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL", "sex", "P4_area", "M1_area",
                   "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$numeric   <- c("NR", "P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL", "sex", "P4_area", "M1_area",
                   "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$categoric <- NULL
crs$target    <- "Sex_Categorical"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- NULL
crs$weights   <- NULL
#============================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=230 train=161 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "sex")
crs$numeric   <- c("P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "sex")
crs$categoric <- NULL
crs$target    <- "JL"
crs$risk      <- NULL
crs$ident     <- "NR"
```

```
crs$ignore    <- c("P4_area", "M1_area", "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio", "M1_L_to_B
crs$weights   <- NULL
#=======================================================================
# Regression model
# Build a Regression model.
crs$glm <- lm(JL ~ ., data=crs$dataset[crs$train,c(crs$input, crs$target)])
# Generate a textual view of the Linear model.
print(summary(crs$glm))
cat('==== ANOVA ====
')
print(anova(crs$glm))
print("
")
# Plot the model evaluation.
ttl <- genPlotTitleCmd("Linear Model",crs$dataname,vector=TRUE)
plot(crs$glm, main=ttl[1])
#=======================================================================
# Evaluate model performance on the full dataset.
# GLM: Generate a Predicted v Observed plot for glm model on UP_for_Jaw.csv.
crs$pr <- predict(crs$glm,
   type    = "response",
   newdata = crs$dataset[c(crs$input, crs$target)])
# Obtain the observed output for the dataset.
obs <- subset(crs$dataset[c(crs$input, crs$target)], select=crs$target)
# Handle in case categoric target treated as numeric.
obs.rownames <- rownames(obs)
obs <- as.numeric(obs[[1]])
obs <- data.frame(JL=obs)
rownames(obs) <- obs.rownames
# Combine the observed values with the predicted.
fitpoints <- na.omit(cbind(obs, Predicted=crs$pr))
# Obtain the pseudo R2 - a correlation.
fitcorr <- format(cor(fitpoints[,1], fitpoints[,2])^2, digits=4)
# Plot settings for the true points and best fit.
op <- par(c(lty="solid", col="blue"))
# Display the observed (X) versus predicted (Y) points.
plot(fitpoints[[1]], fitpoints[[2]], asp=1, xlab="JL", ylab="Predicted")
# Generate a simple linear fit between predicted and observed.
prline <- lm(fitpoints[,2] ~ fitpoints[,1])
# Add the linear fit to the plot.
abline(prline)
# Add a diagonal representing perfect correlation.
par(c(lty="dashed", col="black"))
abline(0, 1)
# Include a pseudo R-square on the plot
legend("bottomright",  sprintf(" Pseudo R-square=%s ", fitcorr),  bty="n")
# Add a title and grid to the plot.
```

```
title(main="Predicted vs. Observed
 Linear Model
 UP_for_Jaw.csv",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["user"]))
grid()
#=====================================================================
building <- TRUE
scoring  <- ! building
# A pre-defined value is used to reset the random seed
# so that results are repeatable.
crv$seed <- 42
#=====================================================================
# Load a dataset from file.
fname         <- "file:///C:/Users/Neda/Desktop/models/DOWN_for_Jaw.csv"
crs$dataset <- read.csv(fname,
na.strings=c(".", "NA", "", "?"),
strip.white=TRUE, encoding="UTF-8")
#=====================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=229 train=160 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("NR", "M1_L", "M1_B", "M2_L", "M2_B", "M3_L",
                   "M3_B", "JL", "sex", "M1_area", "M2_area",
                   "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")
crs$numeric   <- c("NR", "M1_L", "M1_B", "M2_L", "M2_B", "M3_L",
                   "M3_B", "JL", "sex", "M1_area", "M2_area",
                   "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")
crs$categoric <- NULL
crs$target    <- "sex_categorical"
crs$risk      <- NULL
crs$ident     <- NULL
```

```
crs$ignore    <- NULL
crs$weights   <- NULL
#=====================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=229 train=160 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
                   "sex")
crs$numeric   <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
                   "sex")
crs$categoric <- NULL
crs$target    <- "JL"
crs$risk      <- NULL
crs$ident     <- "NR"
crs$ignore    <- c("M1_area", "M2_area", "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio", "M3_L_to_B_
crs$weights   <- NULL
#=====================================================================
# Regression model
# Build a Regression model.
crs$glm <- lm(JL ~ ., data=crs$dataset[crs$train,c(crs$input, crs$target)])
# Generate a textual view of the Linear model.
print(summary(crs$glm))
cat('==== ANOVA ====
')
print(anova(crs$glm))
print("
")
# Plot the model evaluation.
ttl <- genPlotTitleCmd("Linear Model",crs$dataname,vector=TRUE)
plot(crs$glm, main=ttl[1])
#=====================================================================
# Evaluate model performance on the full dataset.
# GLM: Generate a Predicted v Observed plot for glm model on DOWN_for_Jaw.csv.
```

```
crs$pr <- predict(crs$glm,
   type    = "response",
   newdata = crs$dataset[c(crs$input, crs$target)])
# Obtain the observed output for the dataset.
obs <- subset(crs$dataset[c(crs$input, crs$target)], select=crs$target)
# Handle in case categoric target treated as numeric.
obs.rownames <- rownames(obs)
obs <- as.numeric(obs[[1]])
obs <- data.frame(JL=obs)
rownames(obs) <- obs.rownames
# Combine the observed values with the predicted.
fitpoints <- na.omit(cbind(obs, Predicted=crs$pr))
# Obtain the pseudo R2 - a correlation.
fitcorr <- format(cor(fitpoints[,1], fitpoints[,2])^2, digits=4)
# Plot settings for the true points and best fit.
op <- par(c(lty="solid", col="blue"))
# Display the observed (X) versus predicted (Y) points.
plot(fitpoints[[1]], fitpoints[[2]], asp=1, xlab="JL", ylab="Predicted")
# Generate a simple linear fit between predicted and observed.
prline <- lm(fitpoints[,2] ~ fitpoints[,1])
# Add the linear fit to the plot.
abline(prline)
# Add a diagonal representing perfect correlation.
par(c(lty="dashed", col="black"))
abline(0, 1)
# Include a pseudo R-square on the plot
legend("bottomright",  sprintf(" Pseudo R-square=%s ", fitcorr),  bty="n")
# Add a title and grid to the plot.
title(main="Predicted vs. Observed
 Linear Model
 DOWN_for_Jaw.csv",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["user"]))
grid()
#=====================================================================
building <- TRUE
scoring  <- ! building
# A pre-defined value is used to reset the random seed
# so that results are repeatable.
crv$seed <- 42
#=====================================================================
# Load a dataset from file.
fname         <- "file:///C:/Users/Neda/Desktop/models/UP_for_Jaw.csv"
crs$dataset <- read.csv(fname,
na.strings=c(".", "NA", "", "?"),
strip.white=TRUE, encoding="UTF-8")
#=====================================================================
# Action the user selections from the Data tab.
```

```
# Build the train/validate/test datasets.
# nobs=230 train=161 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("NR", "P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL", "sex", "P4_area", "M1_area",
                   "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$numeric   <- c("NR", "P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL", "sex", "P4_area", "M1_area",
                   "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                   "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$categoric <- NULL
crs$target    <- "Sex_Categorical"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- NULL
crs$weights   <- NULL
#============================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=230 train=161 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
```

```
# The following variable selections have been noted.
crs$input     <- c("P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL")
crs$numeric   <- c("P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL")
crs$categoric <- NULL
crs$target    <- "Sex_Categorical"
crs$risk      <- NULL
crs$ident     <- "NR"
crs$ignore    <- c("sex", "P4_area", "M1_area", "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio", "M1
crs$weights   <- NULL
#============================================================================
# Regression model
# Build a Regression model.
crs$glm <- glm(Sex_Categorical ~ .,
    data=crs$dataset[crs$train, c(crs$input, crs$target)],
    family=binomial(link="logit"))
# Generate a textual view of the Linear model.
print(summary(crs$glm))
cat(sprintf("Log likelihood: %.3f (%d df)\n",
            logLik(crs$glm)[1],
            attr(logLik(crs$glm), "df")))
cat(sprintf("Null/Residual deviance difference: %.3f (%d df)\n",
            crs$glm$null.deviance-crs$glm$deviance,
            crs$glm$df.null-crs$glm$df.residual))
cat(sprintf("Chi-square p-value: %.8f\n",
            dchisq(crs$glm$null.deviance-crs$glm$deviance,
                   crs$glm$df.null-crs$glm$df.residual)))
cat(sprintf("Pseudo R-Square (optimistic): %.8f\n",
             cor(crs$glm$y, crs$glm$fitted.values)))
cat('\n==== ANOVA ====\n\n')
print(anova(crs$glm, test="Chisq"))
cat("\n")
# Plot the model evaluation.
ttl <- genPlotTitleCmd("Linear Model",crs$dataname,vector=TRUE)
plot(crs$glm, main=ttl[1])
#============================================================================
# Evaluate model performance on the full dataset.
# Generate an Error Matrix for the Linear model.
# Obtain the response from the Linear model.
crs$pr <- as.vector(ifelse(predict(crs$glm,
   type    = "response",
   newdata = crs$dataset[c(crs$input, crs$target)]) > 0.5, "M", "F"))
# Generate the confusion matrix showing counts.
rattle::errorMatrix(crs$dataset[c(crs$input, crs$target)]$Sex_Categorical, crs$pr, count=TRUE)
# Generate the confusion matrix showing proportions.
(per <- rattle::errorMatrix(crs$dataset[c(crs$input, crs$target)]$Sex_Categorical, crs$pr))
```

```
# Calculate the overall error percentage.
cat(100-sum(diag(per), na.rm=TRUE))
# Calculate the averaged class error percentage.
cat(mean(per[,"Error"], na.rm=TRUE))
#=======================================================================
building <- TRUE
scoring  <- ! building
# A pre-defined value is used to reset the random seed
# so that results are repeatable.
crv$seed <- 42
#=======================================================================
# Load a dataset from file.
fname         <- "file:///C:/Users/Neda/Desktop/models/DOWN_for_Jaw.csv"
crs$dataset <- read.csv(fname,
na.strings=c(".", "NA", "", "?"),
strip.white=TRUE, encoding="UTF-8")
#=======================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=229 train=160 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("NR", "M1_L", "M1_B", "M2_L", "M2_B", "M3_L",
                   "M3_B", "JL", "sex", "M1_area", "M2_area",
                   "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")
crs$numeric   <- c("NR", "M1_L", "M1_B", "M2_L", "M2_B", "M3_L",
                   "M3_B", "JL", "sex", "M1_area", "M2_area",
                   "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio",
                   "M3_L_to_B_ratio")
crs$categoric <- NULL
crs$target    <- "sex_categorical"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- NULL
```

```
crs$weights   <- NULL
#============================================================================
# Action the user selections from the Data tab.
# Build the train/validate/test datasets.
# nobs=229 train=160 validate=34 test=35
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test
# The following variable selections have been noted.
crs$input     <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
                   "JL")
crs$numeric   <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
                   "JL")
crs$categoric <- NULL
crs$target    <- "sex_categorical"
crs$risk      <- NULL
crs$ident     <- "NR"
crs$ignore    <- c("sex", "M1_area", "M2_area", "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio", "M3_
crs$weights   <- NULL
#============================================================================
# Regression model
# Build a Regression model.
crs$glm <- glm(sex_categorical ~ .,
    data=crs$dataset[crs$train, c(crs$input, crs$target)],
    family=binomial(link="logit"))
# Generate a textual view of the Linear model.
print(summary(crs$glm))
cat(sprintf("Log likelihood: %.3f (%d df)\n",
            logLik(crs$glm)[1],
            attr(logLik(crs$glm), "df")))
cat(sprintf("Null/Residual deviance difference: %.3f (%d df)\n",
            crs$glm$null.deviance-crs$glm$deviance,
            crs$glm$df.null-crs$glm$df.residual))
cat(sprintf("Chi-square p-value: %.8f\n",
            dchisq(crs$glm$null.deviance-crs$glm$deviance,
                   crs$glm$df.null-crs$glm$df.residual)))
cat(sprintf("Pseudo R-Square (optimistic): %.8f\n",
```

```
            cor(crs$glm$y, crs$glm$fitted.values)))
cat('\n==== ANOVA ====\n\n')
print(anova(crs$glm, test="Chisq"))
cat("\n")
# Plot the model evaluation.
ttl <- genPlotTitleCmd("Linear Model",crs$dataname,vector=TRUE)
plot(crs$glm, main=ttl[1])
#============================================================
# Evaluate model performance on the full dataset.
# Generate an Error Matrix for the Linear model.
# Obtain the response from the Linear model.
crs$pr <- as.vector(ifelse(predict(crs$glm,
   type    = "response",
   newdata = crs$dataset[c(crs$input, crs$target)]) > 0.5, "M", "F"))
# Generate the confusion matrix showing counts.
rattle::errorMatrix(crs$dataset[c(crs$input, crs$target)]$sex_categorical, crs$pr, count=TRUE)
# Generate the confusion matrix showing proportions.
(per <- rattle::errorMatrix(crs$dataset[c(crs$input, crs$target)]$sex_categorical, crs$pr))
# Calculate the overall error percentage.
cat(100-sum(diag(per), na.rm=TRUE))
# Calculate the averaged class error percentage.
cat(mean(per[,"Error"], na.rm=TRUE))
model1 <- lm(JL ~ P4_L+ P4_B+ P4_GB+ M1_L+ M1_B+ M2_L+ M2_B+sex, data = UP)
vif_values1 <- vif(model1)
print(vif_values1)
model2 <- lm(JL ~ M1_L+ M1_B+ M2_L+ M2_B+ M3_L+ M3_B+sex, data = DOWN)
vif_values2 <- vif(model2)
print(vif_values2)
########################################
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
  crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
  crs$test
# The following variable selections have been noted.
crs$input <- c("NR", "P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
               "M2_L", "M2_B", "JL", "sex", "P4_area", "M1_area",
               "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
```

```
               "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$numeric <- c("NR", "P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                 "M2_L", "M2_B", "JL", "sex", "P4_area", "M1_area",
                 "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio",
                 "M1_L_to_B_ratio", "M2_L_to_B_ratio")
crs$categoric <- NULL
crs$target <- "Sex_Categorical"
crs$risk <- NULL
crs$ident <- NULL
crs$ignore <- NULL
crs$weights <- NULL
#============================================================================
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
  crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
  crs$test
crs$input     <- c("P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL")
crs$numeric   <- c("P4_L", "P4_GB", "P4_B", "M1_L", "M1_B",
                   "M2_L", "M2_B", "JL")
crs$categoric <- NULL
crs$target    <- "Sex_Categorical"
crs$risk      <- NULL
crs$ident     <- "NR"
crs$ignore    <- c("sex", "P4_area", "M1_area", "M2_area", "P4_L_to_GB_ratio", "P4_L_to_B_ratio", "M1
crs$weights   <- NULL
crs$glmUP <- glm(Sex_Categorical ~ .,
                 data=crs$dataset[crs$train, c(crs$input, crs$target)],
                 family=binomial(link="logit"))
vif(crs$glmUP)
#============================================================================
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
```

```
  crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
  crs$test
crs$input     <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
                   "JL")
crs$numeric   <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
                   "JL")
crs$categoric <- NULL
crs$target    <- "sex_categorical"
crs$risk      <- NULL
crs$ident     <- "NR"
crs$ignore    <- c("sex", "M1_area", "M2_area", "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio", "M3_
crs$weights   <- NULL
#============================================================================
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$train <- sample(crs$nobs, 0.7*crs$nobs)
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
  crs$validate
crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
  crs$test
crs$input     <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
                   "sex")
crs$numeric   <- c("M1_L", "M1_B", "M2_L", "M2_B", "M3_L", "M3_B",
                   "sex")
crs$categoric <- NULL
crs$target    <- "JL"
crs$risk      <- NULL
crs$ident     <- "NR"
crs$ignore    <- c("M1_area", "M2_area", "M3_area", "M1_L_to_B_ratio", "M2_L_to_B_ratio", "M3_L_to_B_
crs$weights   <- NULL
#============================================================================
crs$glmDOWN <- glm(sex_categorical ~ .,
                   data=crs$dataset[crs$train, c(crs$input, crs$target)],
                   family=binomial(link="logit"))

vif(crs$glmDOWN)
```