CLASSIFICADOR D'ACCIDENTS GREUS

Ciència de dades per a la Indústria 4.0

9 DE ENERO DE 2023

ETSEIB - UPC
Jan Álvarez Alonso, Ricard Calvo, Steven Chen

Taula de Continguts

1. Resum executiu

La meta de l'estudi actual es analitzar una base de dades on conté informació rellevant d'accidents de Catalunya per aconseguir un model predictiu que pugui explicar si una primera trucada d'emergència es tracta d'un accident molt greu.
Previ al desenvolupament del model, a la *secció 4*, s'analitza i es neteja la informació de la base de dades. S'estudia cada un de les columnes disponibles i es decideix si son rellevants, o si es necessari alguna modificació. S'han realitzat eliminació, combinació i transformacions a les variables convenients.

Un cop treballat les variables disponibles, amb metodologies d'anàlisis de dades es creen diferents models de regressió i models basats en arbres. En concret, s'ha utilitzat: regressió logística, linear *discriminant Analysis* (LDA), *Quadratic Discriminant Analysis* (QDA), *random forest* i *Boosted trees.*

Per avaluar l'efectivitat dels cinc models, s'ha optat comparar el AUC de cadascun dels models. Amb un llindar de 0.17, els millors resultats son els que presenten la regressió logística, linear *discriminant Analysis* (LDA) i *random forest*.

Analitzant el model de regressió logística, s'observa que l'hora en què es produeix un accident influeix de forma notable en la probabilitat que aquest sigui molt greu. En conconcret, un accident ocorregut a les 4 del matí, envers el mateix accident ocorregut a les 12 del migdia té 2,72 vegades més probabilitat de ser molt greu.

Una conclusió similar s'obté en el comportament dels accidents en funció dels mesos, on febrer i maig són els mesos amb un risc més baix, mentre que agost i setembre són els mesos on el mateix accident té una probabilitat més alta de ser molt greu.

El model obtingut ofereix una sensibilitat de 0.689, el que representa que aconseguim identificar el 68,9% dels accidents molt greus. Per un altra banda, el model ofereix una especificitat del 0.682, el que representa que tenim una taxa de falsos positius de 0.318 (Taxa de falsos positius = 1 - 0.682). Això vol dir que identifiquem el 31,8% dels casos greus com a molt greus.

En conclusió, els models obtinguts no son bon predicadors ja que presenten falsos positius i falsos negatius prou significatius que no es poden negligir. Encara així, durant l'anàlisi s'ha pogut conèixer certes tendències i característiques del comportament del model, on recomanen prestar atenció a les hores fosques i els mesos amb més risc de accident greu.

Com a futurs passos d'estudi, es recomana seguir treballant les variables (netejant, eliminant, combinant o transformant) i provar altres mètodes predictius per aconseguir separar de forma més clara les respostes.

2. Objectiu del projecte

L'objectiu del projecte és aconseguir un model que ens ajudi a classificar un accident de trànsit en funció de si és greu o molt greu, a partir d'una sèrie de variables d'entrada que un testimoni de l'accident pugui donar fàcilment. En cas d'obtenir un model fiable, aquest podria ser de gran ajuda per als serveis d'emergència per tenir una primera idea del nombre de recursos que han de mobilitzar i de la urgència amb la que ho han de fer.

A més, amb l'estudi de la gran base de dades de la que es disposa, es pretén identificar els factors de risc que poden afectar en major mesura a la gravetat d'un accident. En cas de trobar aquests, es poden utilitzar per fer campanyes de conscienciació o implementar sistemes per tal d'evitar-los o reduir-los al màxim.

Per últim, es vol analitzar en profunditat el model aconseguit per tal ser conscients de les seves limitacions, i de quina forma es pot millorar les prediccions obtingudes. És vital determinar quines mètriques ens ajudaran a avaluar la fiabilitat del model i com optimitzar-lo.

3. Dades

Les dades utilitzades en aquest projecte han sigut facilitades pel Departament d'Interior de Catalunya i el Servei Català de Trànsit; més concretament obtingudes del portal *dades obertes de Catalunya*. Les dades amb què s'ha treballat han estat actualitzades el 5 de setembre de 2022 per última vegada. Es tracta d'un conjunt de dades estructurat amb 21.161 files i 58 columnes, amb informació relativa als accidents de trànsit amb morts o ferits greus que s'han produït a Catalunya des de l'any 2010. Per cada un dels incidents es té informació del lloc on s'ha produït, la data, l'hora, el nombre i tipus de vehicles implicats, el nombre de víctimes, les condicions meteorològiques, les condicions de la via, i el tipus d'accident, entre d'altres més variables.

Abans de començar l'estudi de les dades, s'ha de netejar i aconseguir donar la forma desitjada al set de dades amb el que es treballa, així com reduir al màxim la seva complexitat. Això implica passos com eliminar casos o variables, adaptar les variables per tal que es pugui treballar correctament amb elles, i realitzar les conversions adients per tal que la resposta obtinguda posteriorment sigui la més precisa possible.

En el moment de començar a tractar les dades s'ha definit el punt de vista que se seguirà per determinar si una variable és útil per l'estudi o no. Es vol utilitzar el model per fer prediccions a partir de les dades proporcionades en una trucada telefònica instants després que succeeixi l'accident. Tenint això present, es realitzen les següents modificacions al set de dades:

- **S'afegeix la columna resposta 'ES_GREU':** Es decideix que la forma de classificar si un accident és o no greu serà a través d'un sistema de puntuació de tal forma que un mort equival a 9 punts, un ferit greu a 3 punts i un ferit lleu a 1 punt. Els incidents que tinguin una puntuació total de >= 9, s'assignaran com a molt greus (=1) i la resta com a greus (=0).
- **S'eliminen les columnes 'F_MORTS', 'F_FERITS_GREUS', 'F_FERITS_LLEUS' i 'F_VICTIMES':** Pel fet que hem fet servir les primeres 3 variables per assignar una puntuació, no té sentit elaborar l'anàlisi de les dades tenint en compte aquestes columnes. La variable 'F_VICTIMES' és la suma de les altres variables, per la qual cosa tampoc ens interessa per l'estudi.
- **S'afegeix la columna 'month':** Es crea la variable 'month' a partir de la data que es dona. Només ens interessa obtenir el mes, ja que el dia de la setmana i el tipus de dia que és, ja és definit per altres variables.
- **S'eliminen les columnes 'Any' i 'dat':** Després d'extreure la informació que ens interessa de la columna 'dat' s'elimina perquè no ens aporta cap valor. De la mateixa forma, l'any de l'accident per si mateix no aporta cap informació significativa per intentar predir la seva gravetat.
- **S'elimina la columna 'D_GRAVETAT':** Es tracta d'una variable que ens dona una resposta similar a la nostra. No és útil utilitzar-la pel nostre model.
- **S'eliminen les columnes 'D_SUBTIPUS_ACCIDENT', 'F_UNITATS_ IMPLICADES', 'pk' i 'F_UNITAT_DESC_IMPLICADES':** 'D_SUBTIPUS_ACCIDENT' és informació similar a la de la columna 'tipAcc' però més detallada. S'ha considerat que en una trucada d'emergència és més rellevant la informació de la columna 'tipAcc' al ser més general (més fàcil de proporcionar) i per tant s'ha eliminat la columna 'D_SUBTIPUS_ACCIDENT'. 'F_UNITATS_IMPLICADES' és la suma de les columnes de vehicles implicats, per tant, té correlació amb aquestes i ens pot portar a errors. El punt quilomètric 'pk' i el subtipus d'accident es considera que no és una informació que sempre es té a l'abast quan ocorre l'accident. Per últim, la darrera variable és sempre 0 pel que tampoc ens aporta cap classe d'informació.

- **S’afegeix la variable ‘hora_punta’:** Veiem útil generar una nova columna segons si l’accident s’ha produït en hora punta o no. Segons dades del RACC, es considera hora punta entre les 7:30 - 9:30 i entre les 18:00 - 20:00 els dies laborables, i entre les 10:00 - 12:00 i les 18:00 - 21:00 quan són dies festius.
- **Es treuen els minuts de la variable ‘hor’:** A l’hora d’analitzar les dades, la hora actuarà com una variable categòrica, pel que no és gaire eficient considerar 1440 categories horàries (24h x 60min). Per aquest motiu es decideix treure els minuts de la variable ‘hor’ i tractar-ho com només 24 categories (una per cada hora).
- **S’elimina la columna "grupHor":** Aquesta columna classifica els accidents en matí, tarda o nit segons l'hora. S’elimina la columna, ja que conté la mateixa informació que “hor” però menys detallada.
- **S’eliminen les columnes ‘via’, ‘nomMun’ i ‘nomCom’:** Aquestes tres variables compten amb 737, 877 i 42 categories respectivament. Això complica molt l’anàlisi pel que es decideix prescindir de les tres. Pel que fa a variables referents a localització, encara es disposa de la demarcació en la variable ‘nomDem’.
- **Es transformen les columnes ‘D_CARRIL_ESPECIAL’, ‘D_CLIMATOLOGIA’, ‘D_FUNC_ESP_VIA’, ‘D_SUPERFICIE’ i ‘D_CIRCULACIO_MESURES_ESP’:** Aquestes variables tenen totes diversos nivells anomenats amb ‘strings’. Se simplifica la variable a només dos nivells (0 i 1) per facilitar l’anàlisi.
- **Es modifica la variable ‘D_LLUMINOSITAT’:** Aquesta variable té 6 categories, cosa que dificulta l’anàlisi. Per simplificar-ho es decideix establir la categoria “De dia, dia clar” com 0, i la resta com a 1. Queden d’aquesta forma només dues categories.
- **Es modifica la variable ‘D_SUPERFICE’:** Aquesta variable té 6 categories. De la mateixa forma que en el cas anterior, ho reduïm a dues de la forma que “Sec i net” és 0 i la resta de categories són 1.
- **S’eliminen les columnes ‘D_INFLUIT_CIRCULACIO’, ’D_INFLUIT_ESTAT_ CLIMA’, ‘D_INFLUIT_LLUMINOSITAT’, ‘D_INFLUIT_MESU_ESP’, ‘D_INFLUIT_ OBJ_CALCADA’, ‘D_INFLUIT_BOIRA’, ‘D_INFLUIT_CARAC_ENTORN’, ‘D_ INFLUIT_INTEN_VENT’, ‘D_INFLUIT_SOLCS_RASES’, ‘D_INFLUIT_ VISIBILITAT’:** S’ha considerat que és poc fiable i molt subjectiu interpretar just després d’un accident quins d’aquests paràmetres han tingut una conseqüència directa en l’accident. A més totes tenen una freqüència molt baixa de ‘sí’, per la qual cosa ens pot esbiaixar el resultat tenint en compte que no sempre tindrem aquestes dades.
- **S’eliminen la columna ‘D_REGULACIO_PRIORITAT’, ‘D_SENTITS_VIA’, ‘D_SUBTIPUS_TRAM’, ‘D_CARACT_ENTORN’, ‘D_TITULARITAT_VIA’, ‘D_TRACTAT_ALTIMETRIC’:** Totes aquestes columnes contenen un gran nombre de ‘NA’. Per facilitar l’anàlisi s’hauria de prescindir de molts casos, pel que és millor eliminar aquestes variables.
- **Es modifiquen totes les variables de tipus ‘char’:** Per tal de poder aplicar els diferents models sense problemes transformem les variables ‘char’ en ‘factor’.
- **S’eliminen les observacions amb NA:** Els NA dificulten molt els anàlisis, pel que es decideix eliminar els casos que en contenen, ja que s’ha comprovat que no suposen una gran quantitat.

Després d’aquestes modificacions ens queda un set de dades amb 17.115 observacions (respecte les 21.161 inicials) i 30 variables (respecte les 58 inicials).
El resultat de les variables netejades es poden visualitzar a la *taula 1*.

Variable	Tipus	Nº niv	Nivells
Zona	Factor	2	"Carretera","Zona urbana"
nomDem	Factor	4	“Barcelona”, “Girona”, “Lleida”, “Tarragona”
F_VIANANTS_IMPLICADES	Int	-	
F_BICICLETES_IMPLICADES	Int	-	
F_CICLOMOTORS_IMPLICADES	Int	-	
F_MOTOCICLETES_IMPLICADES	Int	-	
F_VEH_LLEUGERS_IMPLICADES	Int	-	
F_VEH_PESANTS_IMPLICADES	Int	-	
F_ALTRES_UNIT_IMPLICADES	Int	-	
C_VELOCITAT_VIA	Int	-	
D_ACC_AMB_FUGA	Factor	2	“Si”, “No”
D_BOIRA	Factor	2	“No n'hi ha”, “Si”
D_INTER_SECCIO	Factor	3	“Arribant o eixint intersecció fins 50m”, “Dintre intersecció” , “En secció”
D_LIMIT_VELOCITAT	Factor	2	“Genérica via”, “Senyal velocitat”
D_SUBZONA	Factor	3	“Carretera”, “Travessera”, “Zona urbana”
D_TIPUS_VIA	Factor	6	“Altres”, “Autopista”, “Autovia”, “Camí rural/pista forestal” , “Carretera convencional”, “Via urbana”
grupDiaLab	Factor	2	“CapDeSetmana”, “Feiners”
hor	Factor	24	“0”, ”1”, ”2”, “3”, “4”, “5”, “6”, “7”, “8”, “9”, “10”, “11”, “12”, “13”, “14”, “15”, “16”, “17”, “18”, “19”, “20”, “21”, “22”, “23”, “24”
tipAcc	Factor	6	“Altres”, “Atropellament”, “Bolcada a la calcada”, “Col.lisió d'un vehicle contra un obstacle de la calcada”, “Col.lisió de vehicles en marxa”, “Sortida de la calcada sense especificar”
tipDia	Factor	4	“dg”, “dill-dij”, “dis”, “div”
ES_GREU	num	2	0, 1
hora_punta	Factor	2	“0”, “1”
D_CARRIL_ESPECIAL2	Factor	2	0,1
D_CLIMATOLOGIA2	Factor	2	0,1
D_FUNC_ESP_VIA2	Factor	2	0,1
D_SUPERFICIE2	Factor	2	0,1
D_CIRCULACIO_MESURES_ESP2	Factor	2	0,1
D_LLUMINOSITAT2	Factor	2	0,1
D_VENT2	Factor	2	0,1

Taula 1:Variables netejades

Anàlisi de la multicol·linealitat de les variables numèriques:

El factor d'inflació de la variància (VIF) és una mesura de la multicol·linealitat entre les variables predictores en una regressió múltiple. S'ha calculat el VIF de les variables numèriques de la nostra base de dades *(F_VIANANTS_IMPLICADES + F_BICICLETES_IMPLICADES + F_CICLOMOTORS_IMPLICADES + F_MOTOCICLETES_IMPLICADES + F_VEH_LLEUGERS_IMPLICADES +F_VEH_PESANTS_IMPLICADES + F_ALTRES_UNIT_IMPLICADES+C_VELOCITAT_VIA*).

El resultat (tots les VIFs < 2), ens dona a entendre que no hi ha multicol·linealitat entre aquestes variables.

4. Model

A l'hora d'escollir un model a utilitzar, el primer pas és fixar-se en el tipus de relació que poden mantenir les variables de les que es disposa amb la resposta. Això es pot dur a terme a través de l'anàlisi exploratori o a través d'argumentació logística, si es té prou informació del que descriu cada una de les variables. En el cas que els predictors mantinguin una relació aproximadament lineal, els mètodes clàssics com els de regressió lineal o regressió logística poden donar un resultat més acurat que el mètode d'arbres, el qual no explota les estructures lineals de les diferents variables. En canvi, si les relacions observades són altament no-lineals i complexes, els models basats en arbres poden oferir millor rendiment que els mètodes clàssics mencionats.

En el cas d'estudi, i tal com s'observa a través de l'anàlisi exploratori inicial, resulta difícil establir cap mena de estructura lineal entre les variables, o amb la resposta. Es disposa d'un gran número de possibles predictors i no es poden establir relacions directes, ja que el set de dades conté molta informació i una estructura complexa. Per aquest motiu s'ha decidit estudiar les dades de les que es disposa a través mètodes diferents i comparar-ne els resultats, per tal de decidir posteriorment quin dels models obtinguts s'ajusta millor a la resposta. Els mètodes triats per procedir amb l'anàlisi són el de **regressió logística**, el de ***linear discriminant analysis* (LDA), *Quadratic Discriminant Analysis* (QDA)**, i els mètode basats en arbres de ***random forest* i *Boosted trees***.

Per avaluar l'efectivitat de les diferents metodologies, s'utilitzarà el valor **AUC** màxim com el millor model. A més, donat que es tracta d'accidents greus de trànsit preferim donar falses alarmes a dir que un accident no és molt greu, quan realment ho és. Per aquest motiu, ens fixarem també en la **sensibilitat** dels models, sent millor com més elevat sigui aquest valor. Tot i així, tractarem de mantenir l'**especificitat** el més alt possible, per tal de minimitzar el nombre de vegades que s'actua com si l'accident fos molt greu quan no ho és.

El valor llindar (*threshold*) triat per optimitzar sensibilitat i especificitat és de 0,17. Aquest valor s'ha fixat una vegada obtinguts els models.

4.1 Regressió logística, LDA i QDA

4.1.1 *Stepwise* per seleccionar variables

Donat el gran nombre de variables predictores que tenim en la nostra base de dades s'ha decidit utilitzar la metodologia *Stepwise* per seleccionar-ne només les més significatives.

El resultat del millor model aplicant Stepwise forward. Stepwise backward i *Stepwise both directions* han estat el mateix i és el model que conté les següents variables predictores:

D_SUBZONA + F_VEH_PESANTS_IMPLICADES + F_VEH_LLEUGERS_IMPLICADES + tipAcc + hor + nomDem + C_VELOCITAT_VIA + F_CICLOMOTORS_IMPLICADES + D_TIPUS_VIA + tipDia + F_VIANANTS_IMPLICADES + D_INTER_SECCIO + D_ACC_AMB_FUGA + month + D_SUPERFICIE2 + F_ALTRES_UNIT_IMPLICADES

Observem com hem passat de 29 variables predictores a 16.

A partir de la comparació dels valors AIC dels models de regressió es pot determinar quin és més òptim (com més petit l'AIC, millor model), on per un model logístic considerant totes les variables es **13.860** i per el model obtingut a partir *stepwise* es de de **13.147**.

4.1.2 *5-folds Cross Validation* per avaluar el model.

Per avaluar els models de regressió logística, LDA i QDA s'ha optat per la metodologia k-folds Cross Validation amb k = 5. Això ens permetrà avaluar el model creat a partir d'un set d'entrenament contra un set de test que no haurà vist durant l'entrenament. Existirà una predicció de ES_GREU per cada observació.

4.1.3 *Sensitivity* i *Specificity*

La taula de prediccions dels tres models es poden visualitzar en la *taula 2*.

	Predicted					
	regressioLog		LDA		QDA	
Actual	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE
0	9711	4524	10053	4182	10831	3404
1	897	1983	964	1913	1502	1378

Taula 2: Taula de prediccions dels models de regressió

Els valors de *Sensitivity* i *Specificity* es poden visualitzar a la *taula 3*.

	Sensitivity	**Specificity**
LG	0,689	0,682
LDA	0.665	0.706
QDA	0.478	0.761

Taula 3: Taula de Sensitivitat i especificitat dels models de regressió.

4.2 Mètodes basats en arbres

En el cas dels mètodes basats en arbres no s'ha pre-seleccionat les variables significatives. Tampoc s'ha fet servir la metodologia *k-folds cross validation* i s'ha optat per dividir la base de dades original en una sub-base de dades d'entrenament i una altra de test. En els arbres, per tant, no hi haurà una predicció per cada observació. Només hi haurà prediccions per les observacions catalogades com a test.

Degut al temps de càlcul s'ha fet *Random forest* amb 3000 arbres i *Boosted trees* amb 200 arbres.

4.2.1 *Random Forest*

La taula de prediccions dels tres models es poden visualitzar en la taula 4.

	Predicted	
Actual	FALSE	TRUE

0	2528	1713
1	230	664

Taula 4: Taula de prediccions del model de Random Forest.

Els valors de *Sensitivity* i *Specificity* es poden visualitzar a la *taula 5*.

	Sensitivity	**Specificity**
RF	0.743	0.596

Taula 5: Taula de Sensitivitat i especificitat del model de Random Forest.

Per un altre banda, l'avaluació de la mitjana de decreixement del índex de Gini es pot visualitzar a la figura x, on els nodes son millors i mes purs a mesura que aquest disminueix. Per tant, les variables més rellevants son "hor" i "month".

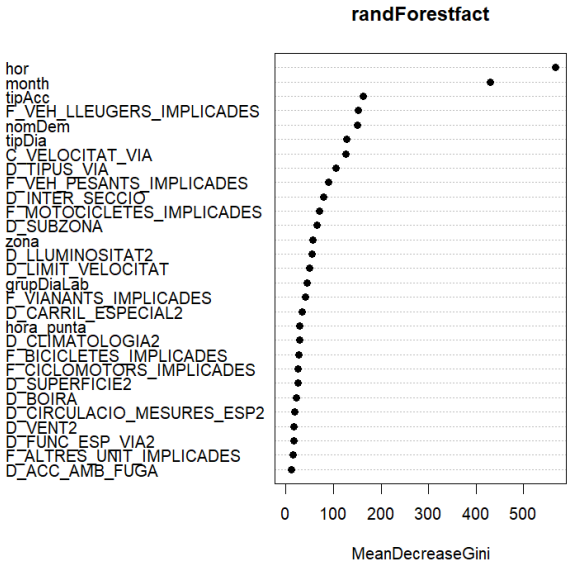

Figura 1: Factors mes importants del model de Random Forest.

4.2.2 Boosted Tree

La taula de prediccions dels tres models es poden visualitzar en la taula 6.

	Predicted	
Actual	FALSE	TRUE
0	2768	1473
1	306	588

Taula 6: Taula de prediccions del model de Boosted Trees.

Els valors de *Sensitivity* i *Specificity* es poden visualitzar a la *taula 7*.

	Sensitivity	**Specificity**
BT	0.743	0.596

Taula 7: Taula de Sensitivitat i especificitat del model de Boosted Trees.

5. Resultats

De tots els models que s'han utilitzat per intentar predir i entendre les dades, es decideix utilitzar el model basat en **regressió logística**. Ens basem en l'AUC com a primer criteri de decisió, i observem que tant el model de regressió logística, el de *Linear Discriminant Analysis* (LDA) i el de *Random Forest* basat en 3000 arbres donen un AUC molt similar, al voltant de **0.732**. Com tots tres prediuen el model amb una precisió similar, ens fixem doncs en les sensibilitats i les especificitats obtingudes en cada model. Amb els llindars proposats (**th = 0,17**), el model de regressió logística és el que ofereix uns valors més elevats i equilibrats a l'hora. A més, aquest resulta ser el model de més fàcil interpretació i del qual podem extreure més informació.

	Logistic	LDA	QDA	Random Forest	Boosted Tree
AUC	0.7319659	0.7327411	0.6836969	0.7310612	0.6913626

Taula 8: Taula de d'AUC dels diferents models.

A la *figura 2* observem que quan ens movem en valors de sensitivitat pròxims al 70% els millors models són el de regressió logística i el LDA.

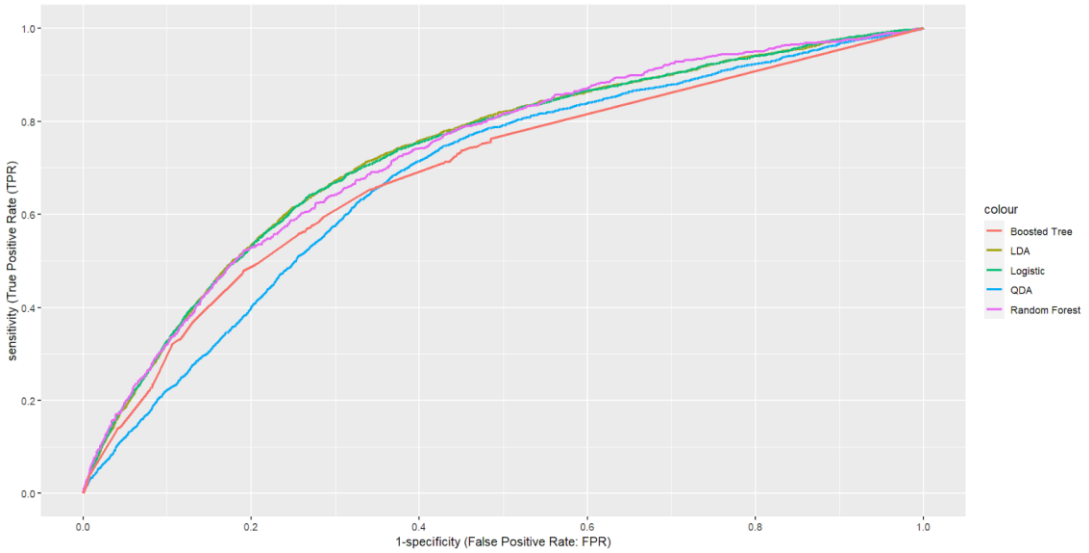

Figura 2: ROC dels diferents models.

El model obtingut, amb les variables que es tenen en compte, així com els seus coeficients és el següent:

log(P(ES_GREU = 1)/ P(ES_GREU = 0)) = -2.37821609320372 + -0.6 * D_SUBZONATravessera + -1.26 * D_SUBZONAZona urbana + 0.79 * F_VEH_PESANTS_IMPLICADES + 0.34 * F_VEH_LLEUGERS_IMPLICADES + -0.05 * tipAccAtropellament + -0.26 * tipAccBolcada a la calcada + 0.34 * tipAccCol.lisió d'un vehicle contra un obstacle de la calcada + -0.09 * tipAccCol.lisió de vehicles en marxa + 0.4 * tipAccSortida de la calcada sense especificar + 0.01 * hor1 + -0.4 * hor10 + -0.58 * hor11 + -0.59 * hor12 + -0.54 * hor13 + -0.35 * hor14 + -0.48 * hor15 + -0.28 * hor16 + -0.5 * hor17 + -0.45 * hor18 + -0.32 * hor19 + 0.07 * hor2 + -0.28 * hor20 + -0.15 * hor21 + 0.12 * hor22 + 0.05 * hor23 + 0.22 * hor3 + 0.41 * hor4 + -0.05 * hor5 + 0.01 * hor6 + -0.4 * hor7 + -0.48 * hor8 + -0.38 * hor9 + 0.33 * nomDemGirona + 0.3 * nomDemLleida + 0.34 * nomDemTarragona + 0.01 * C_VELOCITAT_VIA + -0.75 * F_CICLOMOTORS_IMPLICADES + 0.1 * D_TIPUS_VIAAutopista + 0.07 * D_TIPUS_VIAAutovia + 0 * D_TIPUS_VIACamí rural/pista forestal + 0.42 * D_TIPUS_VIACarretera convencional + 0.64 * D_TIPUS_VIAVia urbana(inclou carrer i carrer residencial) + -0.27 * tipDiadill-dij + -0.05 * tipDiadis + -0.23 * tipDiadiv + 0.31 *

F_VIANANTS_IMPLICADES + -0.02 * D_INTER_SECCIODintre intersecció + 0.18 * D_INTER_SECCIOEn secció + -0.51 * D_ACC_AMB_FUGASi + 0.05 * month10 + -0.01 * month11 + 0.13 * month12 + -0.08 * month2 + 0.05 * month3 + -0.01 * month4 + -0.11 * month5 + -0.05 * month6 + 0.08 * month7 + 0.25 * month8 + 0.23 * month9 + 0.21 * D_SUPERFICIE21 + 0.2 * F_ALTRES_UNIT_IMPLICADES

5.1 Anàlisi del model

Del model obtingut en el punt anterior es pot extreure diversa informació de valor:

- L'hora en què es produeix un accident influeix de forma notable en la probabilitat que aquest sigui molt greu. De fet, es pot veure com entre les 11 i les 14 del migdia és l'hora on un accident té una probabilitat més baixa de ser molt greu, mantenint totes les altres variables constants, mentre que entre les 3 i les 4 de la matinada és l'hora en la qual aquest risc és més alt. Concretament, un accident ocorregut a les 4 del matí, envers el mateix accident ocorregut a les 12 del migdia té 2,72 vegades més probabilitat de ser molt greu. La variable hora, a part, conté informació referent a la lluminositat, columna que ha estat eliminada en fer la metodologia *stepwise*.
  En les tres figures següents es pot comparar com varia aquest factor de risc segons l'hora, la proporció d'accidents registrats segons l'hora i la proporció d'accidents molt greus segons l'hora. Observem que tot i haver molts menys accidents durant la nit (menys circulació), hi ha més probabilitat de patir un accident molt greu.

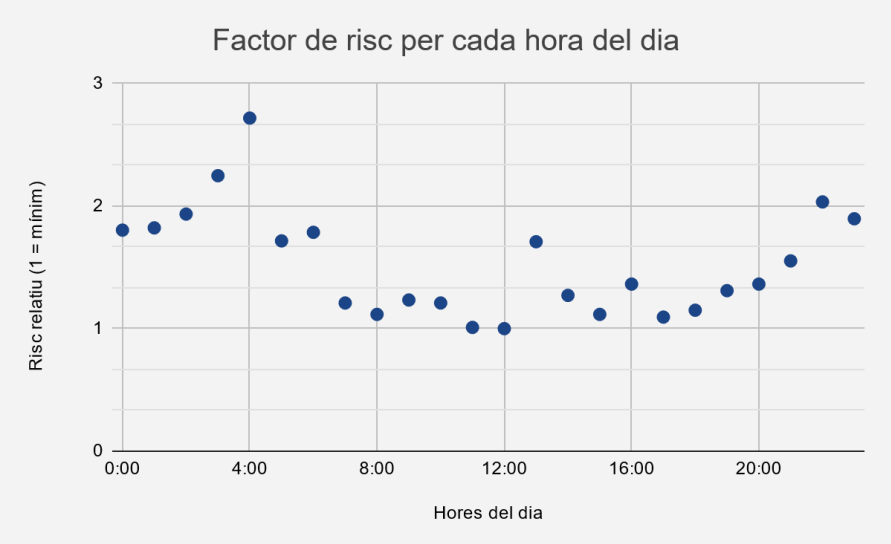

Figura 3: Factor de risc per cada hora del dia.

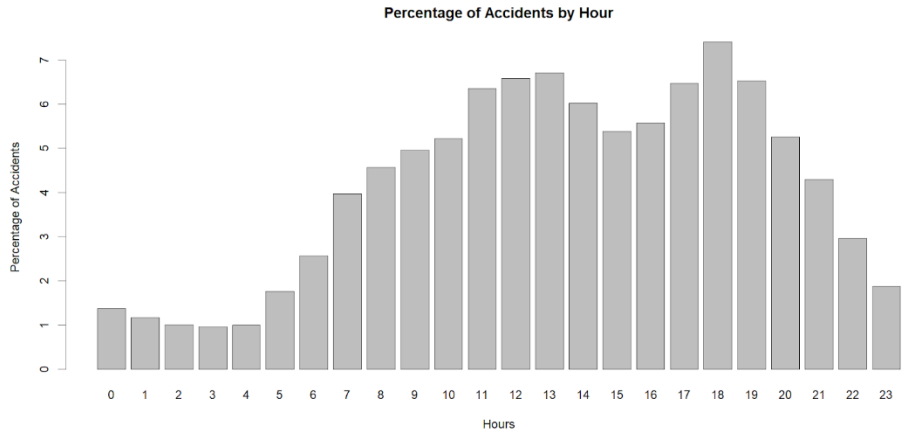

Figura 4: Percentatge d'accidents segons l'hora.

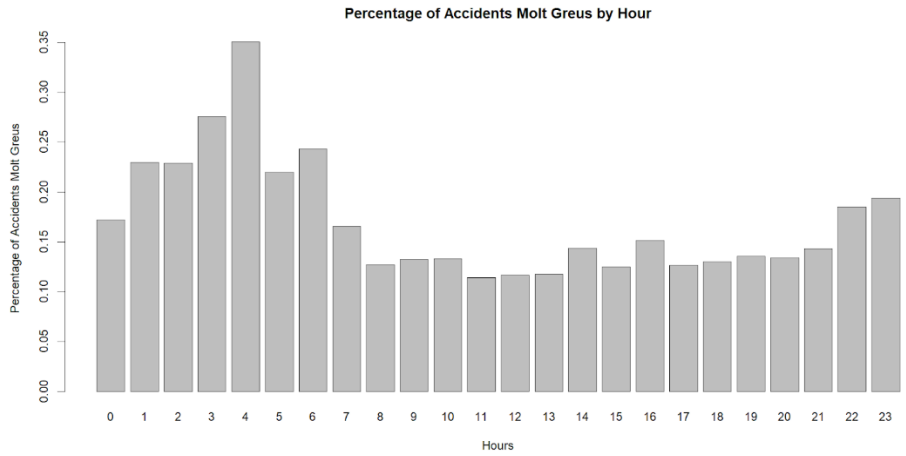

Figura 5: Percentatge d'accidents molt greus segons l'hora.

- La mateixa anàlisi es pot dur a terme pels mesos de l'any, on febrer i maig (*month2* i *month5*) són els mesos amb un risc més baix, mentre que agost i setembre (*month8* i *month9*) són els mesos on el mateix accident té una probabilitat més alta de ser molt greu. Concretament, mantenint totes les variables iguals, un accident ocorregut a l'agost té 1,43 vegades més probabilitat de ser molt greu que en el cas que aquest succeeixi al maig.

  De manera anàloga a les hores, també s'ha comparat el factor de risc, el percentatge d'accidents i el percentatge d'accidents molt greus segons els mesos. En aquest cas, però no s'observa gran diferència pel que fa a quantitat d'accidents ni en la gravetat segons el mes en què es produeix.

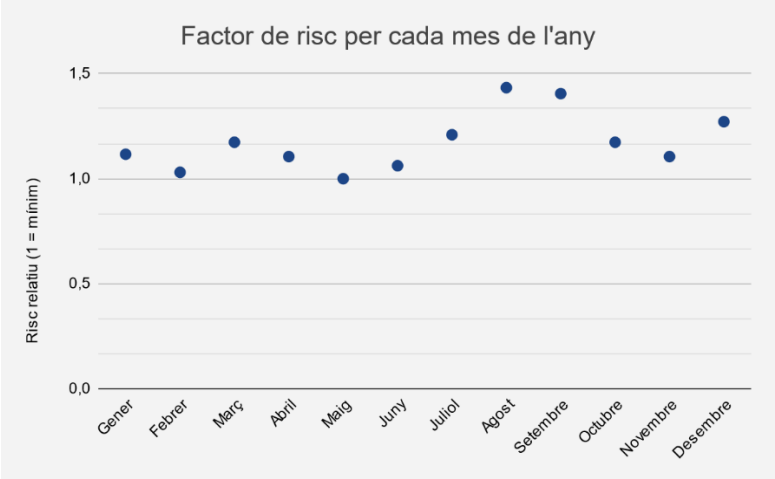

Figura 6: Factor de risc per cada mes de l'any.

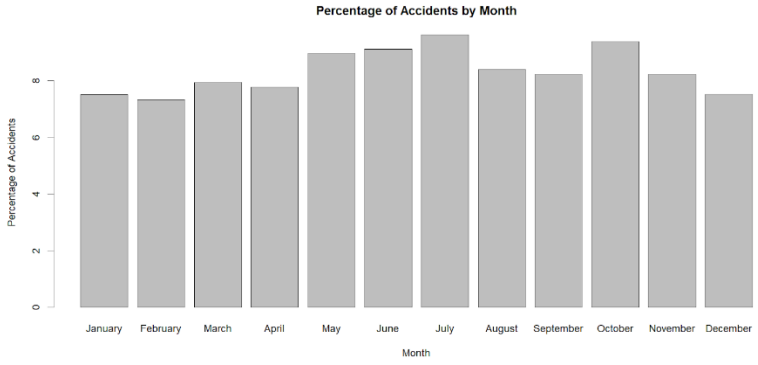

Figura 7: Percentatge d'accidents segons el mes de l'any.

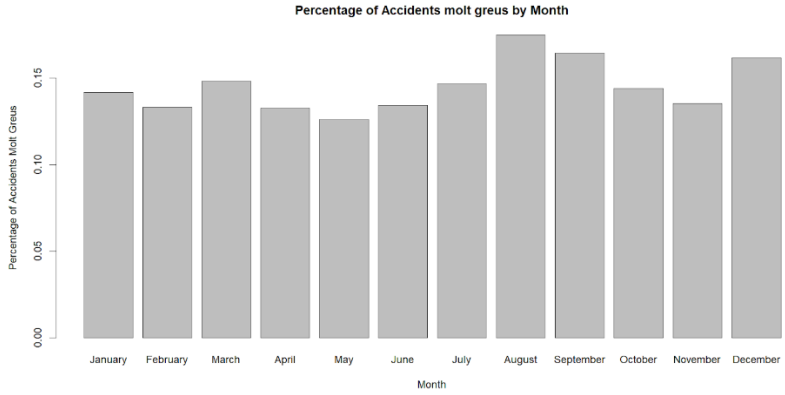

Figura 8: Percentatge d'accidents molt greus segons el mes de l'any.

- Cal tenir en compte també que el valor del coeficient que representa una variable aïllada no sempre es pot relacionar directament amb el seu efecte directe sobre la resposta. Això es deu al fet que algunes d'aquestes variables poden guardar certa correlació, i per això els seus coeficients per si sols, no expliquen completament el seu paper.
- El model obtingut ofereix una sensibilitat de 0.689, el que representa que aconseguim identificar el 68,9% dels accidents molt greus.
- El model obtingut ofereix una especificitat del 0.682, el que representa que tenim una taxa de falsos positius de 0.318 (Taxa de falsos positius = 1 - 0.682). Això vol dir que identifiquem el 31,8% dels casos greus com a molt greus. Com s'ha mencionat, però, això és un punt negatiu que hem d'acceptar si volem tenir una major sensibilitat.

6. Passos següents

En primer lloc, cal dir que el model obtingut no és un bon predictor de la resposta definida. Com s'ha vist en el punt anterior, s'aconsegueixen nivells no negligibles de falsos positius i falsos negatius. És a dir, no s'assoleix separar de forma definida la resposta positiva (molt greu) de la negativa (greu). Per aquest motiu un possible proper pas seria el d'introduir diferents variables al model o utilitzar altres models predictius per aconseguir separar de forma més clara les respostes.

Com s'ha vist en el punt anterior, podem aplicar els resultats obtinguts per fer recomanacions sobre seguretat viària. Tant el model de *Random Forest* com el de regressió logística donen importància a l'hora en què es produeix l'accident. Un missatge per la població podria ser el de prendre màxima precaució i evitar distraccions quan és fosc.

Pel que fa a la informació sobre els mesos de l'any, ens pot servir per incrementar la publicitat sobre seguretat viària en els mesos de més incidència pel que fa a la gravetat dels accidents (agost i setembre).

Si s'hagués de tornar a fer el projecte des de 0 caldria recollir les dades tal com s'han acabat transformant en el codi, així disminuir el temps de neteja de dades. És a dir, intentant minimitzar les categories en moltes de les variables predictives. També seria interessant estipular una forma de catalogar els accidents segons la gravetat de forma sistemàtica.

7. Annex (not included in the page count)

R Code

```
# Treball del Curs Data Science MUEI

# Students: Jan Álvarez Alonso, Ricard Calvo, Steven Chen

# 09-01-2023


# 1. Introducció ----------------------------------------------------------------


# Loading Libraries


library(ggplot2)

library(tidyverse)

library(leaps) # needed for the best subset modeling

library(MASS) # lda, qda functions

library(ROCR) #to compute the AUC

library(stargazer)

library(dplyr)

library(lubridate)

library(tree)

library(randomForest) # bagging and random forest

library(gbm) # boosting

library(caTools)

library(car)

library(pROC)

require(pROC)
```

```
# Loading Data


dforiginal <- read.csv2(file =
'Accidents_de_tr_nsit_amb_morts_o_ferits_greus_a_Catalunya.csv', sep=',')

accidents <- read.csv2(file =
'Accidents_de_tr_nsit_amb_morts_o_ferits_greus_a_Catalunya.csv', sep=',')



#Canviem els "sense especificar" per NA

accidents[accidents == "Sense Especificar"] <- NA

accidents[accidents == "Sense especificar"] <- NA



# 2. Afegir/eliminar/nateja de dades ----------------------------------------




# Add column of accident Score:  9 x F_MORTS + 3 x F_GREUS + 1 x F_LLEUS



accidents$accident_Score <-
9*accidents$F_MORTS+3*accidents$F_FERITS_GREUS+accidents$F_FERITS_LLE
US



# Add column "ES_GREU". Aquesta serà la variable resposta a predir

accidents$ES_GREU <- ifelse(accidents$accident_Score >= 9, 1, 0)



table(accidents$ES_GREU)



# Add column of month. A partir de la data creem una columna dels mesos



accidents$month <- month(accidents$dat)
```

```
accidents$month <- as.character(accidents$month)

# Add column of day time slot.
# A partir de la hora creem columna de les hores com a categories

accidents$hor <- as.character(as.integer(accidents$hor))

# Eliminem la columna "grupHor" ja que està correlacionada amb "hor"

accidents<-subset(accidents,select = -c(grupHor))

# Add column that defines if it's a high-traffic hour or not

accidents$hora_punta <- as.factor(ifelse(
  # high-traffic hours are from 7:30 to 9:30 and 18:00 to 20:00 during work days
  (accidents$grupDiaLab %in% c("Feiner")) &
    ((accidents$hor >= "7.30" & accidents$hor <= "9.30") |
       (accidents$hor >= "18.00" & accidents$hor <= "20.00")),
  1,  # high-traffic hour
  # high-traffic hours are from 10:00 to 12:00 and 18:00 to 21:00 on weekends
  ifelse(
    (accidents$grupDiaLab %in% c("CapDeSetmana", "Festiu")) &
      ((accidents$hor >= "10.00" & accidents$hor <= "12.00") |
         (accidents$hor >= "18:00" & accidents$hor <= "21:00")),
    1,  # high-traffic hour
    0  # not a high-traffic hour
```

```
  )
))


# Delete of columns
# S'eliminen columnes que considrem inneceçàries per l'anàlisi
accidents<-subset(accidents,select = -c(Any,dat, pk, F_MORTS, F_FERITS_GREUS,
                                        F_FERITS_LLEUS, F_VICTIMES, D_GRAVETAT,
                                        D_SUBTIPUS_ACCIDENT, accident_Score,
                                        F_UNIT_DESC_IMPLICADES))



# A partir d'aquí mirarem les seguents columnes:
# via, nomMun,
# D_CARRIL_ESPECIAL,D_CLIMATOLOGIA,
# D_FUNC_ESP_VIA,D_INFLUIT_SOLCS_RASES,
# D_INTER_SECCIO,D_SUPERFICIE,nomCom,
# D_ACC_AMB_FUGA,D_CIRCULACIO_MESURES_ESP,
# D_INFLUIT_CIRCULACIO,D_INFLUIT_ESTAT_CLIMA,
# D_INFLUIT_LLUMINOSITAT, D_INFLUIT_MESU_ESP,
# D_INFLUIT_OBJ_CALCADA,
# D_LLUMINOSITAT,
# D_REGULACIO_PRIORITAT, D_SENTITS_VIA,
# D_SUBTIPUS_TRAM, D_VENT
# Aquestes columnes tenen el problema de contenir moltes subcategories.
```

```
## Eliminem columna "via" perq te 737 categories

length(levels(as.factor(accidents$via)))

accidents<-subset(accidents,select = -c(via))


## Eliminem columna "nomMun" perq te 877 categories

length(levels(as.factor(accidents$nomMun)))

accidents<-subset(accidents,select = -c(nomMun))


## Eliminem columna "nomCom" perq te 42 categories

table(accidents$nomCom, useNA = "always")

length(levels(as.factor(accidents$nomCom)))

accidents<-subset(accidents,select = -c(nomCom))



## F_UNITATS_IMPLICADES

# Eliminem la columna F_UNITATS_IMPLICADES, perquè és la suma de

# les següents columnes


accidents<-subset(accidents,select = -c(F_UNITATS_IMPLICADES))



## D_CARRIL_ESPECIAL

table(accidents$D_CARRIL_ESPECIAL, useNA = "always")

# Add column of D_CARRIL_ESPECIAL2 (1 si n'hi ha , 0 si no hi ha)

accidents$D_CARRIL_ESPECIAL2 <-
as.factor(ifelse(accidents$D_CARRIL_ESPECIAL == "No n'hi ha", 0, 1))
```

```
# Eliminem files NA de D_CARRIL_ESPECIAL2

accidents <-accidents %>% drop_na(D_CARRIL_ESPECIAL2)


# Eliminem la columna original D_CARRIL_ESPECIAL

accidents<-subset(accidents,select = -c(D_CARRIL_ESPECIAL))




## D_CLIMATOLOGIA

table(accidents$D_CLIMATOLOGIA, useNA = "always")


# Eliminem files NA

accidents <-accidents %>% drop_na(D_CLIMATOLOGIA)


# Canviem les categories a Bon temps==0 mal temps ==1

accidents$D_CLIMATOLOGIA2 <- as.factor(ifelse(accidents$D_CLIMATOLOGIA ==
"Bon temps",

                                0, 1))


# Eliminem la columna original D_CLIMATOLOGIA

accidents<-subset(accidents,select = -c(D_CLIMATOLOGIA))

table(accidents$D_CLIMATOLOGIA2, useNA = "always")




## D_FUNC_ESP_VIA

table(accidents$D_FUNC_ESP_VIA, useNA = "always")


# Eliminem files NA
```

```
accidents <-accidents %>% drop_na(D_FUNC_ESP_VIA)


# Canviem les categories a Sense funció especial==0,  la resta==1

accidents$D_FUNC_ESP_VIA2 <- as.factor(ifelse(accidents$D_FUNC_ESP_VIA ==
"Sense funció especial",

                                0, 1))

# Eliminem la columna original D_FUNC_ESP_VIA

accidents<-subset(accidents,select = -c(D_FUNC_ESP_VIA))

table(accidents$D_FUNC_ESP_VIA2, useNA = "always")




## D_INFLUIT_SOLCS_RASES

table(accidents$D_INFLUIT_SOLCS_RASES, useNA = "always")


# Eliminem files NA

accidents <-accidents %>% drop_na(D_INFLUIT_SOLCS_RASES)




## D_INTER_SECCIO

table(accidents$D_INTER_SECCIO, useNA = "always")




## D_SUPERFICIE

table(accidents$D_SUPERFICIE, useNA = "always")


# Canviem les categories a Sec i net==0,  la resta==1
```

```
accidents$D_SUPERFICIE2 <- as.factor(ifelse(accidents$D_SUPERFICIE == "Sec i
net",

                                    0, 1))


# Eliminem la columna original D_SUPERFICIE

accidents<-subset(accidents,select = -c(D_SUPERFICIE))

table(accidents$D_SUPERFICIE2, useNA = "always")




## D_ACC_AMB_FUGA

table(accidents$D_ACC_AMB_FUGA, useNA = "always")


# Eliminem files NA

accidents <-accidents %>% drop_na(D_ACC_AMB_FUGA)




## D_CIRCULACIO_MESURES_ESP

table(accidents$D_CIRCULACIO_MESURES_ESP, useNA = "always")


# Eliminem files NA

accidents <-accidents %>% drop_na(D_CIRCULACIO_MESURES_ESP)


# Canviem les categories a No n'hi ha==0,  la resta==1

accidents$D_CIRCULACIO_MESURES_ESP2 <-
as.factor(ifelse(accidents$D_CIRCULACIO_MESURES_ESP == "No n'hi ha",

                                    0, 1))
```

```
# Eliminem la columna original D_CIRCULACIO_MESURES_ESP

accidents<-subset(accidents,select = -c(D_CIRCULACIO_MESURES_ESP))

table(accidents$D_CIRCULACIO_MESURES_ESP2, useNA = "always")



## D_INFLUIT_CIRCULACIO

table(accidents$D_INFLUIT_CIRCULACIO, useNA = "always")



## D_INFLUIT_ESTAT_CLIMA

table(accidents$D_INFLUIT_ESTAT_CLIMA, useNA = "always")



## D_INFLUIT_LLUMINOSITAT

table(accidents$D_INFLUIT_LLUMINOSITAT, useNA = "always")



## D_INFLUIT_MESU_ESP

table(accidents$D_INFLUIT_MESU_ESP, useNA = "always")



## D_INFLUIT_OBJ_CALCADA

table(accidents$D_INFLUIT_OBJ_CALCADA, useNA = "always")

# Observem com:

# D_INFLUIT_CIRCULACIO,D_INFLUIT_ESTAT_CLIMA,
```

```
# D_INFLUIT_LLUMINOSITAT, D_INFLUIT_MESU_ESP,

# D_INFLUIT_OBJ_CALCADA,

# Tenen una freqüencia de si molt baixa, decidim eliminar aquestes columnes


accidents<-subset(accidents,select = -
c(D_INFLUIT_CIRCULACIO,D_INFLUIT_ESTAT_CLIMA,

                                D_INFLUIT_LLUMINOSITAT, D_INFLUIT_MESU_ESP,

                                D_INFLUIT_OBJ_CALCADA))


## D_LLUMINOSITAT

table(accidents$D_LLUMINOSITAT, useNA = "always")


# Hi ha 6 categories

prop.table(table(accidents$D_LLUMINOSITAT,accidents$ES_GREU),1)

prop.table(table(accidents$D_LLUMINOSITAT,accidents$ES_GREU),2)


# Decidim crear una nova columna nomes amb 2 categories. 0 =="De dia, dia clar"

# 1 == la resta


accidents$D_LLUMINOSITAT2 <- as.factor(ifelse(accidents$D_LLUMINOSITAT ==
"De dia, dia clar",

                              0, 1))


# Eliminem la columna original D_LLUMINOSITAT

accidents<-subset(accidents,select = -c(D_LLUMINOSITAT))

table(accidents$D_LLUMINOSITAT2, useNA = "always")

prop.table(table(accidents$D_LLUMINOSITAT2,accidents$ES_GREU),1)
```

```
prop.table(table(accidents$D_LLUMINOSITAT2,accidents$ES_GREU),2)
```

```
## D_REGULACIO_PRIORITAT
table(accidents$D_REGULACIO_PRIORITAT, useNA = "always")

# Moltes NA, eliminem la columna
accidents<-subset(accidents,select = -c(D_REGULACIO_PRIORITAT))
```

```
## D_SENTITS_VIA
table(accidents$D_SENTITS_VIA, useNA = "always")

# Moltes NA, eliminem columna
accidents<-subset(accidents,select = -c(D_SENTITS_VIA))
```

```
## D_SUBTIPUS_TRAM
table(accidents$D_SUBTIPUS_TRAM, useNA = "always")

# Moltes NA, eliminem columna
accidents<-subset(accidents,select = -c(D_SUBTIPUS_TRAM))
```

```
## D_VENT
table(accidents$D_VENT, useNA = "always")
```

```
# Decidim crear una nova columna nomes amb 2 categories. 0=="Calma, vent molt
suau"

# 1 == la resta

accidents$D_VENT2 <- as.factor(ifelse(accidents$D_VENT == "Calma, vent molt
suau",

                                      0, 1))



# Eliminem la columna original D_VENT

accidents<-subset(accidents,select = -c(D_VENT))

table(accidents$D_VENT2, useNA = "always")




## D_ACC_AMB_FUGA

table(accidents$D_ACC_AMB_FUGA, useNA = "always")




## D_BOIRA

table(accidents$D_BOIRA, useNA = "always")




## D_CARACT_ENTORN

table(accidents$D_CARACT_ENTORN, useNA = "always")


# Moltes NA, eliminem columna

accidents<-subset(accidents,select = -c(D_CARACT_ENTORN))
```

```
## D_INFLUIT_BOIRA

table(accidents$D_INFLUIT_BOIRA, useNA = "always")

# Moltes NA, eliminem columna

accidents<-subset(accidents,select = -c(D_INFLUIT_BOIRA))

## D_INFLUIT_CARACT_ENTORN

table(accidents$D_INFLUIT_CARACT_ENTORN, useNA = "always")

# Moltes NA (comparats amb "SI"), eliminem columna

accidents<-subset(accidents,select = -c(D_INFLUIT_CARACT_ENTORN))

## D_INFLUIT_INTEN_VENT

table(accidents$D_INFLUIT_INTEN_VENT, useNA = "always")

# Moltes NA (comparats amb "SI"), eliminem columna

accidents<-subset(accidents,select = -c(D_INFLUIT_INTEN_VENT))

## D_INFLUIT_SOLCS_RASES

table(accidents$D_INFLUIT_SOLCS_RASES, useNA = "always")

# Pocs "SI", eliminem columna
```

```
accidents<-subset(accidents,select = -c(D_INFLUIT_SOLCS_RASES))



## D_INFLUIT_VISIBILITAT

table(accidents$D_INFLUIT_VISIBILITAT, useNA = "always")

# Moltes NA (comparats amb "SI"), eliminem columna

accidents<-subset(accidents,select = -c(D_INFLUIT_VISIBILITAT))



## D_TITULARITAT_VIA

table(accidents$D_TITULARITAT_VIA, useNA = "always")

# Moltes NA, eliminem columna

accidents<-subset(accidents,select = -c(D_TITULARITAT_VIA))



## D_TRACAT_ALTIMETRIC

table(accidents$D_TRACAT_ALTIMETRIC, useNA = "always")

# Moltes NA, eliminem columna

accidents<-subset(accidents,select = -c(D_TRACAT_ALTIMETRIC))



## C_VELOCITAT_VIA

table(accidents$C_VELOCITAT_VIA, useNA = "always")
```

```
# Eliminem files NA

accidents <-accidents %>% drop_na(C_VELOCITAT_VIA)


# Eliminem les files amb valor =999 (deu ser un error)

accidents <- accidents[accidents$C_VELOCITAT_VIA != 999, ]




## Canviem les columnes amb chr datatype a factor datatype


accidents[sapply(accidents, is.character)] <- lapply(accidents[sapply(accidents,
is.character)],

                                       as.factor)


str(accidents)




# 3. Anàlisi de les dades ----------------------------------------------------


summary(accidents)

head(accidents)


# Mirem si hi ha relació entre les variables numèriques


modelLogSencer <- glm(ES_GREU ~ F_VIANANTS_IMPLICADES

                      + F_BICICLETES_IMPLICADES + F_CICLOMOTORS_IMPLICADES
```

```
                + F_MOTOCICLETES_IMPLICADES +
F_VEH_LLEUGERS_IMPLICADES

                +F_VEH_PESANTS_IMPLICADES + F_ALTRES_UNIT_IMPLICADES+

                  C_VELOCITAT_VIA,

                family = 'binomial', data = accidents)

vif(modelLogSencer)


# Com no hi ha cap VIF>5 totes les variables númeriques son linaement indep




# 4. Stepwise approach per seleccionar variables ------------------------------


# null model


modelNull <- glm(ES_GREU ~ 1, data = accidents)


# full model


modelFull <- glm(ES_GREU ~ .,data = accidents)


# 4.1. Stepwise forward -------------------------------------------------------


modSWFwd <-  step(modelNull,

              scope=list(lower=modelNull,upper=modelFull),

              direction='forward',

              trace=1)
```

```
# 4.2. Stepwise backward ----------------------------------------------------


modSWBwd <-  step(modelFull,

                  scope=list(lower=modelNull,upper=modelFull),

                  direction='backward',

                  trace=1)


# 4.3. Stepwise both directions ---------------------------------------------


modSWBoth <-  step(glm(ES_GREU ~ nomDem + F_BICICLETES_IMPLICADES +

                       F_CICLOMOTORS_IMPLICADES +
F_MOTOCICLETES_IMPLICADES + F_VEH_PESANTS_IMPLICADES +

                       C_VELOCITAT_VIA + D_ACC_AMB_FUGA + D_INTER_SECCIO,
data = accidents),

                   scope=list(lower=modelNull,upper=modelFull),

                   direction='both',

                   trace=1)


lapply(list(modSWFwd, modSWBwd, modSWBoth), FUN = summary)


# Surten el mateix model amb el mateix AIC = 13147


# 4.4. Best glm model -------------------------------------------------------


modelSW <- glm(formula = ES_GREU ~ D_SUBZONA +
F_VEH_PESANTS_IMPLICADES +

    F_VEH_LLEUGERS_IMPLICADES + tipAcc + hor + nomDem +
C_VELOCITAT_VIA +
```

```
    F_CICLOMOTORS_IMPLICADES + D_TIPUS_VIA + tipDia +
F_VIANANTS_IMPLICADES +

    D_INTER_SECCIO + D_ACC_AMB_FUGA + month + D_SUPERFICIE2 +

    F_ALTRES_UNIT_IMPLICADES, data = accidents)



modelFull <- glm(ES_GREU ~ .,

            family = 'binomial', data = accidents)


lapply(list(modelSW = modelSW, modelFull = modelFull),extractAIC)



# 5. CV - Five Folds ----------------------------------------------------------


# Canviem nom a la base de dades

df <- accidents


# Canviar nom columna resposta!

colnames(df)[which(names(df) == "ES_GREU")] <- "resposta"

set.seed(1964) # Llavor

nfolds=5

nObs <- nrow(df) # Numero observacions

folds <- sample(nfolds, size = nObs, replace = T)

df$fold <- folds


table(df$fold, df$resposta)
```

```
# 5.1 Five folds (accurate)  (o n folds) ---------------------------------------

ngreu<-filter(df, resposta==0)

greu<-filter(df, resposta==1)


set.seed(1963)


# assign a fold to ngreu cases

tamanypaquet=trunc(nrow(ngreu)/nfolds)

tamanypaquetultim=(nrow(ngreu))-(nfolds-1)*tamanypaquet

nB = c(rep(1:(nfolds-1),each=tamanypaquet),rep(nfolds,tamanypaquetultim))

nB = sample(nB)

ngreu$fold = nB


# assign a fold to greu cases

tamanypaquet2=trunc(nrow(greu)/nfolds)

tamanypaquetultim2=(nrow(greu))-(nfolds-1)*tamanypaquet2

nM = c(rep(1:(nfolds-1),each=tamanypaquet2),rep(nfolds,tamanypaquetultim2))

nM = sample(nM)

greu$fold = nM


# rejoin data

df <- rbind(ngreu,greu)

table(df$fold, df$resposta)



# 6. 5-CV amb ModelSW ----------------------------------------------------------
```

```
# definim threshold

th = 0.17

# 6.1 Regressio Logistica ----------------------------------------------------

df$logisticPred<-1

for(fold in 1:nfolds) {

  modelLog <- glm(resposta ~ D_SUBZONA + F_VEH_PESANTS_IMPLICADES +
                  F_VEH_LLEUGERS_IMPLICADES + tipAcc + hor + nomDem +
C_VELOCITAT_VIA +
                  F_CICLOMOTORS_IMPLICADES + D_TIPUS_VIA + tipDia +
F_VIANANTS_IMPLICADES +
                  D_INTER_SECCIO + D_ACC_AMB_FUGA + month + D_SUPERFICIE2 +
                  F_ALTRES_UNIT_IMPLICADES,
                family = 'binomial', data = df[df$fold != fold,])

  predLog <- predict(modelLog, newdata = df[df$fold == fold,], type = 'response')

  df$logisticPred[df$fold == fold] <- predLog

}

# 6.2. Lineal Discriminant Analysis ------------------------------------------
```

```
df$ldaPred<-1

for(fold in 1:nfolds) {

  modLda = lda(resposta ~ D_SUBZONA + F_VEH_PESANTS_IMPLICADES +
                 F_VEH_LLEUGERS_IMPLICADES + tipAcc + hor + nomDem +
C_VELOCITAT_VIA +
                 F_CICLOMOTORS_IMPLICADES + D_TIPUS_VIA + tipDia +
F_VIANANTS_IMPLICADES +
                 D_INTER_SECCIO + D_ACC_AMB_FUGA + month + D_SUPERFICIE2 +
                 F_ALTRES_UNIT_IMPLICADES,
               data = df[df$fold != fold, 1:30])

  predLda = predict(modLda,
                    newdata = df[df$fold == fold, ])

  df$ldaPred[df$fold == fold] <- predLda$posterior[,2]

}

# 6.3. Quadratic Discriminant Analysis ----------------------------------------

df$qdaPred<-1

for(fold in 1:nfolds) {

  modQda = qda(resposta ~ D_SUBZONA + F_VEH_PESANTS_IMPLICADES +
```

```
            F_VEH_LLEUGERS_IMPLICADES + tipAcc + hor + nomDem +
C_VELOCITAT_VIA +

            F_CICLOMOTORS_IMPLICADES + D_TIPUS_VIA + tipDia +
F_VIANANTS_IMPLICADES +

            D_INTER_SECCIO + D_ACC_AMB_FUGA + month + D_SUPERFICIE2 +

            F_ALTRES_UNIT_IMPLICADES,

          data = df[df$fold != fold, 1:30])


  predQda = predict(modQda,

              newdata = df[df$fold == fold, ])


  df$qdaPred[df$fold == fold] <- predQda$posterior[,2]


}



# 7 Comparar els 3 MODELS LINEALS 5-V ModelSW ----------------------------------

# 7.1 Matriu confusio regressioLog ---------------------------------------------


nObs <- nrow(df)

taulaLog <- table(df$resposta, df$logisticPred>th, dnn = c('actual','predicted'))

print(taulaLog)


# Error rate es fa servir quan els falsos negatius i falsos positius tenen costos similars

ErrorRateLG= (taulaLog[2]+taulaLog[3])/nObs

ErrorRateLG
```

```
# Accuracy (igual q error rate pero mires el encert) es fa servir quan els falsos negatius
i falsos positius tenen costos similars

AccuracyLG = 1-ErrorRateLG

AccuracyLG



# Sensitivitat es fa servir quan no acceptes falsos negatius. Ex: cancer o Covid

SensitivityLG = taulaLog[4]/(taulaLog[2]+taulaLog[4])

SensitivityLG



# Specificity quan no vols donar falses alarmes

SpecificityLG = taulaLog[1]/(taulaLog[1]+taulaLog[3])

SpecificityLG



# Precission quan vols minimitzar falsos positius (Ex:spam)

PrecissionLG = taulaLog[4]/(taulaLog[3]+taulaLog[4])

PrecissionLG



# 7.2 Matriu confusio LDA ----------------------------------------------------



taulaLDA <- table(df$resposta, df$ldaPred>th, dnn = c('actual','predicted'))

print(taulaLDA)



# Error rate es fa servir quan els falsos negatius i falsos positius tenen costos similars

ErrorRateLDA= (taulaLDA[2]+taulaLDA[3])/nObs

ErrorRateLDA
```

```
# Accuracy (igual q error rate pero mires el encert) es fa servir quan els falsos negatius
i falsos positius tenen costos similars

AccuracyLDA = 1-ErrorRateLDA

AccuracyLDA


# Sensitivitat es fa servir quan no acceptes falsos negatius. Ex: cancer o Covid

SensitivityLDA = taulaLDA[4]/(taulaLDA[2]+taulaLDA[4])

SensitivityLDA


# Specificity quan no vols donar falses alarmes

SpecificityLDA = taulaLDA[1]/(taulaLDA[1]+taulaLDA[3])

SpecificityLDA


# Precission quan vols minimitzar falsos positius (Ex:spam)

PrecissionLDA = taulaLDA[4]/(taulaLDA[3]+taulaLDA[4])

PrecissionLDA


# 7.3 Matriu confusio QDA ----------------------------------------------------


taulaQDA <- table(df$resposta, df$qdaPred>th, dnn = c('actual','predicted'))

print(taulaQDA)


# Error rate es fa servir quan els falsos negatius i falsos positius tenen costos similars

ErrorRateQDA= (taulaQDA[2]+taulaQDA[3])/nObs

ErrorRateQDA
```

```
# Accuracy (igual q error rate pero mires el encert) es fa servir quan els falsos negatius i falsos positius tenen costos similars

AccuracyQDA = 1-ErrorRateQDA

AccuracyQDA


# Sensitivitat es fa servir quan no acceptes falsos negatius. Ex: cancer o Covid

SensitivityQDA = taulaQDA[4]/(taulaQDA[2]+taulaQDA[4])

SensitivityQDA


# Specificity quan no vols donar falses alarmes

SpecificityQDA = taulaQDA[1]/(taulaQDA[1]+taulaQDA[3])

SpecificityQDA


# Precission quan vols minimitzar falsos positius (Ex:spam)

PrecissionQDA = taulaQDA[4]/(taulaQDA[3]+taulaQDA[4])

PrecissionQDA



# 7.4 AUC dels 3 models lineals ----------------------------------------------


predROClog <- prediction(predictions = df[,'logisticPred'],labels = df$resposta)

predROClda <- prediction(predictions = df[,'ldaPred'],labels = df$resposta)

predROCqda <- prediction(predictions = df[,'qdaPred'],labels = df$resposta)


perfROClog <- performance(prediction.obj = predROClog, measure = 'auc')

perfROClda <- performance(prediction.obj = predROClda, measure = 'auc')

perfROCqda <- performance(prediction.obj = predROCqda, measure = 'auc')
```

```
perfROClog@y.values[[1]]


# AUC = 0.7319659


perfROClda@y.values[[1]]


# AUC = 0.7327411


perfROCqda@y.values[[1]]


# AUC = 0.6836969


# ROC curves


ROC <- list(performance(prediction.obj = predROClog, measure = 'tpr',x.measure =
'fpr'),
            performance(prediction.obj = predROClda, measure = 'tpr',x.measure = 'fpr'),
            performance(prediction.obj = predROCqda, measure = 'tpr',x.measure = 'fpr'))


ggplot(data = NULL,aes(x=x,y=y)) +
  geom_step(data = data.frame(x = ROC[[2]]@x.values[[1]],
                              y = ROC[[2]]@y.values[[1]]), aes(col='LDA'), direction = 'vh', size =
1) +
  geom_step(data = data.frame(x = ROC[[3]]@x.values[[1]],
                              y = ROC[[3]]@y.values[[1]]), aes(col='QDA'), direction = 'vh', size =
1) +
  geom_step(data = data.frame(x = ROC[[1]]@x.values[[1]],
```

```
                    y = ROC[[1]]@y.values[[1]]), aes(col='Logistic'), direction = 'vh', size
= 1) +

  labs(x='1-specificity (False Positive Rate: FPR)',y='sensitivity (True Positive Rate
(TPR)') +

  scale_x_continuous(breaks=seq(0,1,0.2)) +

  scale_y_continuous(breaks=seq(0,1,0.2))




# 8. Tree based methods ----------------------------------------------------

# 8.1 Random forest (factors) ----------------------------------------------


# 8.1.1 Splitting data in train and test data ------------------------------


set.seed(120)


trainSize = round(dim(accidents)[1]*0.7,0)

train = sample(1:dim(accidents)[1], trainSize)

test = -train


# 8.1.2 Fitting Random Forest to the train dataset -------------------------

set.seed(120)  # Setting seed


randForestfact = randomForest(as.factor(ES_GREU) ~ .,

                    data = accidents[train,],

                    ntree = 3000,

                    mtry = round((sqrt(ncol(accidents)-1))))
```

```
# 8.1.3 Predicting the Test set results ----------------------------------------

RFfactPred = predict(randForestfact, newdata = accidents[test,], type='prob')



RFfactPred <- (RFfactPred[,2])



taulaRF = table(accidents[test,]$ES_GREU, RFfactPred > th,dnn =
c('actual','predicted'))

print(taulaRF)



# 8.1.4 Plotting model ---------------------------------------------------------

plot(randForestfact)

varImpPlot(randForestfact, pch = 16)



# Error rate es fa servir quan els falsos negatius i falsos positius tenen costos similars

ErrorRateRF= (taulaRF[2]+taulaRF[3])/nObs

ErrorRateRF



# Accuracy (igual q error rate pero mires el encert) es fa servir quan els falsos negatius
i falsos positius tenen costos similars

AccuracyRF = 1-ErrorRateRF

AccuracyRF



# Sensitivitat es fa servir quan no acceptes falsos negatius. Ex: cancer o Covid

SensitivityRF = taulaRF[4]/(taulaRF[2]+taulaRF[4])

SensitivityRF



# Specificity quan no vols donar falses alarmes
```

```
SpecificityRF = taulaRF[1]/(taulaRF[1]+taulaRF[3])

SpecificityRF



# Precission quan vols minimitzar falsos positius (Ex:spam)

PrecissionRF = taulaRF[4]/(taulaRF[3]+taulaRF[4])

PrecissionRF



# 8.2 AUC Random Forest factors ----------------------------------------------



predROCRFfact <- prediction(predictions =
as.numeric(as.character(RFfactPred)),labels = accidents[test,]$ES_GREU)



perfROCRFfact <- performance(prediction.obj = predROCRFfact, measure = 'auc')



perfROCRFfact@y.values[[1]]



# AUC = 0.7310612



ROC <- list(performance(prediction.obj = predROClog, measure = 'tpr',x.measure =
'fpr'),

          performance(prediction.obj = predROClda, measure = 'tpr',x.measure = 'fpr'),

          performance(prediction.obj = predROCqda, measure = 'tpr',x.measure = 'fpr'),

          performance(prediction.obj = predROCRFfact, measure = 'tpr',x.measure =
'fpr'))



ggplot(data = NULL,aes(x=x,y=y)) +

  geom_line(data = data.frame(x = ROC[[2]]@x.values[[1]],

                              y = ROC[[2]]@y.values[[1]]), aes(col='LDA'), size = 1) +
```

```
  geom_line(data = data.frame(x = ROC[[3]]@x.values[[1]],
                              y = ROC[[3]]@y.values[[1]]), aes(col='QDA'), size = 1) +
  geom_line(data = data.frame(x = ROC[[1]]@x.values[[1]],
                              y = ROC[[1]]@y.values[[1]]), aes(col='Logistic'), size = 1) +
  geom_line(data = data.frame(x = ROC[[4]]@x.values[[1]],
                              y = ROC[[4]]@y.values[[1]]), aes(col='Random Forest Factor'), size
= 1)+
  labs(x='1-specificity (False Positive Rate: FPR)',y='sensitivity (True Positive Rate
(TPR)') +
  scale_x_continuous(breaks=seq(0,1,0.2)) +
  scale_y_continuous(breaks=seq(0,1,0.2))


# 9. Boosted tree ------------------------------------------------------------


set.seed(120)  # Setting seed
boostedTree = gbm(ES_GREU ~ .,
                  data = accidents[train,],
                  distribution = 'bernoulli',         #Bernuilli perq és de dos factors 0 i 1
                  shrinkage = 0.001,
                  n.trees = 200,
                  interaction.depth = 2)


summary(boostedTree, main = 'Boosted tree (B = 200; d = 3; lambda = 0.01)')


gbmPred <- predict(boostedTree, newdata = accidents[test,], type = 'response')


# Confusion matrix
```

```
taulaBT = table(accidents[test,]$ES_GREU, gbmPred > th,dnn = c('actual','predicted'))

print(taulaBT)


# Error rate es fa servir quan els falsos negatius i falsos positius tenen costos similars

ErrorRateBT= (taulaBT[2]+taulaBT[3])/nObs

ErrorRateBT


# Accuracy (igual q error rate pero mires el encert) es fa servir quan els falsos negatius
i falsos positius tenen costos similars

AccuracyBT = 1-ErrorRateBT

AccuracyBT


# Sensitivitat es fa servir quan no acceptes falsos negatius. Ex: cancer o Covid

SensitivityBT = taulaBT[4]/(taulaBT[2]+taulaBT[4])

SensitivityBT


# Specificity quan no vols donar falses alarmes

SpecificityBT = taulaBT[1]/(taulaBT[1]+taulaBT[3])

SpecificityBT


# Precission quan vols minimitzar falsos positius (Ex:spam)

PrecissionBT = taulaBT[4]/(taulaBT[3]+taulaBT[4])

PrecissionBT


# 9.1 AUC Boosted tree ----------------------------------------------------------


predROCbt <- prediction(predictions = gbmPred,labels = accidents[test,]$ES_GREU)
```

```
perfROCbt <- performance(prediction.obj = predROCbt, measure = 'auc')


perfROCbt@y.values[[1]]

#AUC = 0.6913626




# 10. ROC dels 5 models ----------------------------------------------------


ROC <- list(performance(prediction.obj = predROClog, measure = 'tpr',x.measure =
'fpr'),

          performance(prediction.obj = predROClda, measure = 'tpr',x.measure = 'fpr'),

          performance(prediction.obj = predROCqda, measure = 'tpr',x.measure = 'fpr'),

          performance(prediction.obj = predROCRFfact, measure = 'tpr',x.measure =
'fpr'),

          performance(prediction.obj = predROCbt, measure = 'tpr',x.measure = 'fpr'))


ggplot(data = NULL,aes(x=x,y=y)) +

  geom_line(data = data.frame(x = ROC[[2]]@x.values[[1]],

                              y = ROC[[2]]@y.values[[1]]), aes(col='LDA'), size = 1) +

  geom_line(data = data.frame(x = ROC[[3]]@x.values[[1]],

                              y = ROC[[3]]@y.values[[1]]), aes(col='QDA'), size = 1) +

  geom_line(data = data.frame(x = ROC[[1]]@x.values[[1]],

                              y = ROC[[1]]@y.values[[1]]), aes(col='Logistic'), size = 1) +

  geom_line(data = data.frame(x = ROC[[4]]@x.values[[1]],

                              y = ROC[[4]]@y.values[[1]]), aes(col='Random Forest'), size = 1)+

  geom_line(data = data.frame(x = ROC[[5]]@x.values[[1]],
```

```
                    y = ROC[[5]]@y.values[[1]]), aes(col='Boosted Tree'), size = 1)+

  labs(x='1-specificity (False Positive Rate: FPR)',y='sensitivity (True Positive Rate
(TPR)') +

  scale_x_continuous(breaks=seq(0,1,0.2)) +

  scale_y_continuous(breaks=seq(0,1,0.2))




# 11. Analitzar el model ----------------------------------------------------



#model final:



modelLogFinal <- glm(ES_GREU ~ D_SUBZONA + F_VEH_PESANTS_IMPLICADES
+

                F_VEH_LLEUGERS_IMPLICADES + tipAcc + hor + nomDem +
C_VELOCITAT_VIA +

                F_CICLOMOTORS_IMPLICADES + D_TIPUS_VIA + tipDia +
F_VIANANTS_IMPLICADES +

                D_INTER_SECCIO + D_ACC_AMB_FUGA + month + D_SUPERFICIE2 +

                F_ALTRES_UNIT_IMPLICADES,

              family = 'binomial', data = accidents)




summary(modelLog)

summary(modelLog)$coef



# 11.1 Escrivim eqüació logOdds ---------------------------------------------



# Extract the coefficients
```

```
coefficients <- modelLogFinal$coefficients

# Get the names of the variables
variable_names <- names(coefficients)

# The intercept is the first element in the coefficients vector
intercept <- coefficients[1]

# Initialize an empty equation
equation <- ""

# Loop through the rest of the coefficients
for(i in 2:length(coefficients)) {
  # Round the coefficient to the nearest cent
  coefficient <- round(coefficients[i], 2)

  # Add the term for this variable to the equation
  equation <- paste(equation, "+", coefficient, "*", variable_names[i])
}

# Prefix the intercept
equation <- paste("log(P(ES_GREU = 1)/ P(ES_GREU = 0)) =", intercept, equation)

# Print the equation
print(equation)
```

```
# 12. Plot Graphics ------------------------------------------------------------


# Count the number of accidents for each month

accidents_by_month <- table(accidents$month)


# Calculate the percentage of accidents for each month

accidents_by_month_percent <- (accidents_by_month / sum(accidents_by_month)) *
100


# Create a bar chart

barplot(accidents_by_month_percent[order(as.numeric(names(accidents_by_month_p
ercent)))],

        main = "Percentage of Accidents by Month",

        xlab = "Month",

        ylab = "Percentage of Accidents",

        names.arg = c("January", "February", "March", "April", "May", "June", "July",
"August", "September", "October", "November", "December"))


# Count the number of accidents for each month

accidents_by_hor <- table(accidents$hor)


# Calculate the percentage of accidents for each hor

accidents_by_hor_percent <- (accidents_by_hor / sum(accidents_by_hor)) * 100


# Create a bar chart

barplot(accidents_by_hor_percent[order(as.numeric(names(accidents_by_hor_percent
)))],

        main = "Percentage of Accidents by Hour",
```

```
        xlab = "Hours",

        ylab = "Percentage of Accidents")



# Count the number of accidents for each month

accidents_by_month <- table(accidents$month)



# Calculate the percentage of accidents for each month

accidents_by_month_percent <- prop.table(table(accidents$ES_GREU,
accidents$month),2)[2,]



# Create a bar chart

barplot(accidents_by_month_percent[order(as.numeric(names(accidents_by_month_p
ercent)))],

        main = "Percentage of Accidents molt greus by Month",

        xlab = "Month",

        ylab = "Percentage of Accidents Molt Greus",

        names.arg = c("January", "February", "March", "April", "May", "June", "July",
"August", "September", "October", "November", "December"))



# Count the number of accidents for each month

accidents_by_hor <- table(accidents$hor)



# Calculate the percentage of accidents for each hor

accidents_by_hor_percent <- prop.table(table(accidents$ES_GREU,
accidents$hor),2)[2,]



# Create a bar chart

barplot(accidents_by_hor_percent[order(as.numeric(names(accidents_by_hor_percent
)))],
```

```
main = "Percentage of Accidents Molt Greus by Hour",
xlab = "Hours",
ylab = "Percentage of Accidents Molt Greus")
```