Universitat Politècnica de Catalunya

Facultat d'Informàtica de Barcelona
Facultat de Matemàtiques i Estadística
Escola Tècnica Superior d'Enginyeria de Telecomunicació

Grau en Ciència i Enginyeria de Dades

Aprenentatge Automàtic 1

Predicció de la gravetat dels accidents: Un enfocament de l'aprenentatge automàtic per millorar la seguretat viària

---

Autors: Miquel Roca i Pol Resina

Índex

1 Introducció

1.1 Motivació

Any rere any, milions de vides es veuen afectdes pels accidents de trànsit arreu del món. No només s'han de tenir en compte les pèrdudes físiques, sino que també la congestió del trànsit, provocant retencions, desviaments i talls de carreteres. La congestió flueix per les carreteres, agreujant els temps de viatge. Els viatgers experimenten frustració, pèrdua de productivitat i possibles impactes econòmics a causa del retard. De la mateixa manera, l'augment del risc d'accidents posteriors agreuja la situació.

Per això, degut a aquesta problemàtica hem decidit desenvolupar una eina d'aprenentatge automàtic per predir la gravetat dels accidents al trànsit donades unes condicions inicials així com la temperatura o la presència de senyals de trànsit a prop d'on ha sigut l'accident.

1.2 Visió General del Projecte

El *dataset* escollit correspon a accidents de trànsit que cobreix 49 estats dels Estats Units. Segons l'autor [1] de la base de dades, aquestes es recullen contínuament a partir del febrer de 2016. Per dur a terme la recollida de dades, s'han utilitzat diversos proveïdors, incloent diverses API que proporcionen dades a temps real. Aquestes transmeten esdeveniments de trànsit capturats per a diferents entitats importants així com el departament de transport, les agències d'aplicació de la llei, càmeres de trànsit i sensors dins de les xarxes de carreteres. L'autor proporciona dos *datasets*: El *full* que correspon a un amb un total de 7,7 milions d'observacions i un *sampled* amb 500k observacions. Tal i com indica l'autor, aquesta partició s'agafa de l'original *dataset* amb valors aleatoris. Degut a la capacitat en memòria[2] i de les dimensions del *dataset* original, per aquest treball hem decidit treballar a partir del *dataset sampled*[3].

Així doncs, la variable a predir o *target* per aquest treball serà la *Severity*. Aquesta mostra la gravetat de l'accident que correspon a un número entre 1 i 4, on 1 indica el menor impacte en el trànsit. El Departament de Transport dels Estats Units té publicat quatre classes de gravetat per a accidents en transports de motor. Aquestes son els següents, en ordre ascendent: accidents amb danys al vehicle, accidents amb lesions lleus, accidents no fatals i accidents fatals.[4] Potser és interessant tenir això en compte a l'hora de seleccionar les mètriques de validació.

Finalment, estaria bé destacar que l'objectiu d'aquest informe és explicar el perquè de les decisions que s'han anat fent. Els *notebooks* i codis proporcionats estan suficientment explicats como per entendre el què s'ha fet, però les decisions s'explicaran en aquest *paper* principalment. Aquesta mateixa idea s'anirà seguint durant tot el *report*.

2 Exploració de Dades

És interessant destacar que es tracta d'un problema de classificació ja que la variable a predir està formada per quatre classes.

En aquesta primera secció, hem decidit fer una visió general de les dades abans de començar el preprocessament. Totes les explicacions i primeres observacions es poden veure al *notebook*

[1] Per a més informació sobre l'autor i en concret la base de dades, premeu aquí. De la mateixa manera, el *paper* de com s'ha fet la base de dades el podeu trobar aquí.

[2] Fins i tot no tenim suficient memòria RAM per carregar en memòria totes les dades

[3] S'ha de tenir en compte ja que de certa menera s'estaria esbiaixant les dades

[4] Aquestes dades es poden consultar aquí.

anomenat *data_exploration*.

En primer lloc, hem començat a dividir les variables segons el seu tipus i secció. Per exemple, les variables de la carretera de l'accident són booleanas i fan referència al mateix tipus.

- Variables a predir: *Severity*
- Variables de localització: *Street*, *City*, *County*, *State*, *Zipcode*, *Country*, *Start_Lat*, *Start_Lng*, *End_Lat*, *End_Lng*, *Airport_Code*, *Timezone*
- Variables temporals: *Start_Time*, *End_Time*, *Weather_Timestamp*
- Variables de trànsit: *ID*, *Source*, *Description*, *Distance(mi)*
- Variables de clima: *Temperature(F)*, *Wind_Chill(F)*, *Humidity(%)*, *Pressure(in)*, *Visibility(mi)*, *Wind_Direction*, *Wind_Speed*, *Precipitation*, *Weather_Condition*
- Variables de la carretera: *Amenity*, *Bump*, *Crossing*, *Give_Away*, *Junction*, *No_Exit*, *Railway*, *Roundabout*, *Station*, *Stop*, *Traffic_Calming*, *Traffic_Signal*, *Turning_Loop*
- Variables de periodes del dia: *Sunrise_Signal*, *Civil_Twilight*, *Nautical_Twilight*, *Astronomical_Twilight*

Una vegada hem observat per primera vegada les dades, el més interessant a destacar és el desbalanceig de la variable a predir. En la Figura 1, podem veure com hi ha poques dades per a valors extrems. Això ens fa pensar que mètriques com l'*accuracy* no seran d'utilitat i haurem d'utilitzar el *F-score* o haurem de balancejar. No obstant això, a primera vista podem pensar que balancejar dades no acaba de ser bona idea ja que ens quedariem amb només 12000 observacions que corresponen a un 2,5% de les dades originals.

A nivell general, gràcies a aquesta primera tasca, s'ha entès el comportament de totes les variables. Per exemple, a la Figura 2 es mostra la procedència dels accidents utilitzant la longitud i la latitud, que com podem veure no hi ha cap valor atípic ja que es pot veure el mapa dels EEUU d'Amèrica sense cap mena d'error. De la mateixa manera, les variables de la carretera i de periodes del dia son *booleanes* i predominen *Fals* i *Day*, respectivament. Les variables de trànsit son numèriques i el comportament és diferent pel tipus de variable a analitzar. Finalment, és interessant destacar que hem desglossat la variable *Start_Time* en messos, dies de la setmana i anys per veure quan s'han produit els accidents on hem vist resultats coherents.

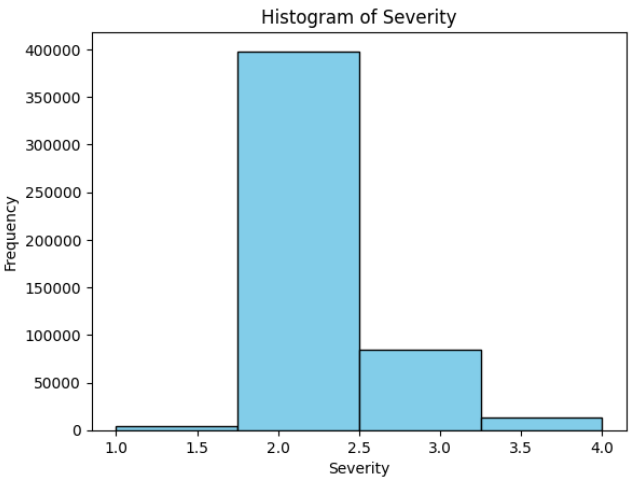

Figura 1: Histograma de la variable *Severity*.

Figura 2: Representació de les latituds i longituds dels accidents.

3 Preprocessament

Degut al gran volum de dades que hi ha, fer un bon preprocesament és una tasca necessaria per obtenir bones prediccions i resultats. Per això, es llisten els passos que s'han decidit fer pel que fa referència al preprocessament. Aquests mateixes seccions es poden trobar al *notebook* de preprocesament.

En general, hem seguit un preprocessament de les dades estàndard: primera visió de les dades, treballar amb outliers i amb *missing values*, *feature engineering*, normalització d'algunes dades i finalment una conclusió i visió general del preprocessament. No obstant això, hem decidit fer dos canvis.

En primer lloc, fer dues fases de sel·lució de característiques degut al gran volum de dades que tenim. Abans de tractar amb valors atípics i que falten, hem decidit eliminar un conjunt de característiques que no creiem que son rellevants per predir la variable *target, Severity*. Això no només facilita la comprensió del *dataset* ja que redueix la dimensionalitat, sino que també facilita altres tasques com ara la simplificació de càlculs posteriors com la imputació de valors que falten, per exemple.

En segon lloc, també hem decidit fer el *feature engineering* abans de detectar els valors atípics i els *missing values* ja que les característiques generades a través de l'extracció aporten informació valuosa que podria influir en la segona ronda de selecció de característiques. Això ens serà d'utilitat per evaluar si realment expliquen la variable a predir o no.

Una vegada aclarit això, es procedeix a llistar tots els passos que hem dut a terme.

1. *First Feature selection*: Primera selecció prèvia d'atributs importants.
2. *Feature extraction*: Transformar dades originals en representacions més compactes.
3. *Dealing with missing values*: Estratègies per gestionar dades nul·les
4. *Dealing with outliers*: Mètodes per identificar i gestionar dades inusuals.
5. *Second Feature selection*: Escollir els atributs importants per fer prediccions.
6. *Normalization*: Escalar dades a un rang estàndard per fer una comparació i anàlisis consistent.
7. *Ending the preprocessing*: Finalització de la manipulació de dades abans d'entrenar el model.

3.1 *First Feature selection*

Aquesta tasca és molt important degut a que tenim 46 variables d'entrada on hem de treure algunes per tal d'evitar *overfitting* i estalviar-nos costos de computació. Decidim treure aquestes variables: *ID*, *Source*, *Description*, *Street*, *City*, *County*, *State*, *Zipcode*, *Country*, *Timezone*, *Weather_Timestamp*, *End_Lat*, *End_Lng*, *Airport_Code*, *Wind_Chill(F)*, *Wind_Direction*, *Wind_Speed(mph)*, *Precipitation(in)*, *Weather_Condition*, *Bump*, *Sunrise_Sunset*, *Civil_Twilight*, *Nautical_Twilight*, *Astronomical_Twilight*.

Les raons van des de que creiem que no són importants a l'hora de predir la Severity, com és el cas de *ID* o *Description*. Algunes són redundants, com *End_Lat* i *End_Lng*, ja que són molt similars a l'inici. I altres perquè creiem que no són tan rellevants com les que hem deixat. Finalment, també hauriem de considerar les variables que no aporten més informació en el cas de la localització. Una vegada tenim la longitud i la latitud, no és d'utilitat saber altres com ara el carrer o la ciutat, per exemple. Estariem donant al model la mateixa informació.

3.2 *Feature extraction*

L'extracció de característiques a partir de variables transforma dades existents en altres per millorar models, revelar patrons i augmentar la precisió predictiva. En el nostre cas, les variables temporals *Start_Time* i *End_Time* es codifiquen com una *string*, i això mateix ens causa problemes a l'hora d'imputar valors amb l'algorisme *KNN* on $k = 1$ per reduir el cost computacional. Hem decidit crear les següents variables: *Time_Difference*, *Year*, *Month*, *Day*, *Hour*, *Weekday*. Finalment, eliminem les dues variables originals les quals hem utilitzat per crear les anteriors.

3.3 *Dealing with missing values*

Observant el diagrama de barres de la Figura 3, veiem que el propi programa detecta ja 4 variables que tenen Nan's. A part d'això, observarem les variables numèriques una per una per tal de veure si n'hi ha valors que s'han codificat d'una altra manera i són Nan's o si són valors que s'han ficat per ficar, com per exemple succeix a la variable temperatura on tenim alguna observació amb temperatures que són impossibles d'observar a la Terra. Anem pas per pas.

De cada variable numèrica, hem observat els seus valors més petits i més grans i hem decidit si eren missing values o no. De distància no és possible que la longitud sigui de 0, ja que com a mínim un petit tram de la carretera es veu afectat per aquest. De temperatura el comentat anteriorment, als EEUU és impossible observar temperatures de -40F o de 135F, són massa extrems. De humitat valors de l'1 o del 100 per cent són exagerats, ja que estaríem parlant de climes extrems. A la pressió, obervem que els valors més petits de 20 són valors que no es poden donar a la Terra. De visibilitat tenim valors de 0, els quals són impossibles ja que es veu alguna cosa, i els valors més grans de 30 milles també són impossibles ja que no podem arribar a veure tant lluny.

Amb aquestes classificiacions, el que hem fet és donar valor Nan a les files d'aquestes columnes que satisfaguin les condicions esmenades. A la Figura 4, observem el canvi en els Nan's després de fer aquesta tasca. Fem això per després utilitzant l'algoritme de KNN amb un $k = 1$, un veí de proximitat, imputar els valors que són Nan en el nostre *dataset*. Segons si la variable que té missing values és numèrica o categòrica, cal fer servir el KNNRegressor o el KNNClassifier. Per variables de tipus string provoca errors i és més complicat, per això hem eliminat les variables de tipus string anteriorment. Un cop estimats aquests valors, els fiquem al *dataset* i així obtenim 500000 observacions sense cap missing value.

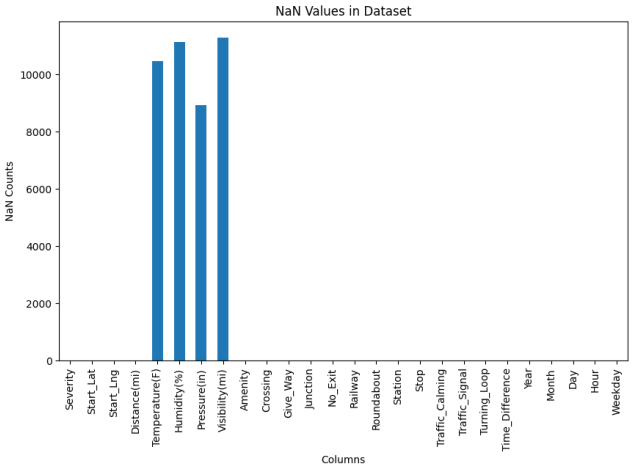

Figura 3: Diagrama de barres dels Nan's abans inspeccionar Nan's.

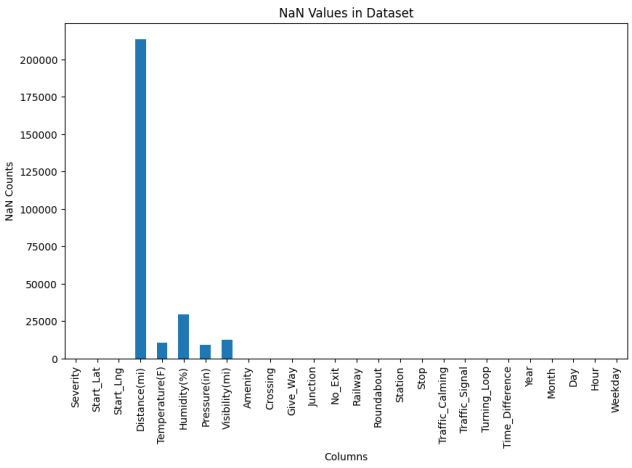

Figura 4: Diagrama de barres dels Nan's després inspeccionar Nan's.

3.4 *Dealing with outliers*

Un cop tractats els valors anteriors, hem de tractar els valors atípics. Aquests són valors que no representen el conjunt de mostres i no són gaire significatius. També produeixen errors en certs models els quals són molt sensibles a aquests valors, ja que a l'hora d'ajustar les dades tenen en compte aquestes obervacions i no obtenim bons resultats. Les estratègies que ens hem plantejat han sigut dues: IQR i LocalOutlierFactor. Per la primera variable vam probar els dos mètodes i vam veure que el segon mètode trigava més i donava pitjors resultats que l'IQR, per tant hem procedit amb aquest. L'IQR treu les observacions que estan per sota i per sobre del Q1 Q3, respectivament, 1,5 cops la diferència entre el Q1 i el Q3. Aquest procediment l'hem fet per cada variable la qual observàvem que tenia valors atípics. La decisió de si hi havia atípics l'hem feta amb l'histograma i el diagrama de caixa i bigotis. Si hi ha valors que sobresurten molt d'on la gran majoria de valors hi eren, hem aplicat el procediment de l'IQR ja que és un mètode clàssic i molt sòlid.

Hem hagut de treure atípics de pràcticament totes les columnes, excepte d'algunes com les de la posició d'inici, ja que creiem que com cada accident ha sigut en un lloc únic, en coordenades, no podem dir que aquests valors siguin observacions molt extranyes. De Temperatura, Distància, Visibilitat, Pressió i Time Difference. En les Figures 5 i 6 observem que hem millorat molt la distribució de la variable Distància aplicant la tècnica de l'IQR.

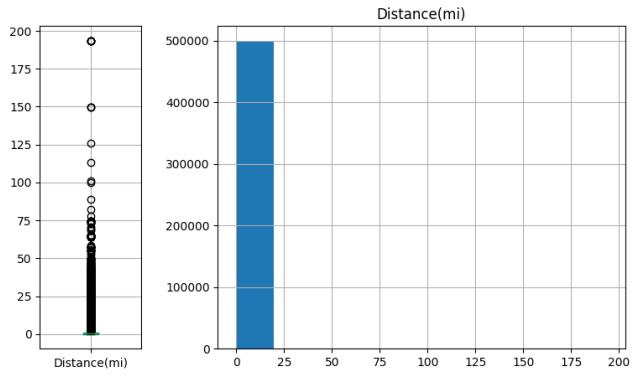

Figura 5: Boxplot i Histograma de Distància amb atípics.

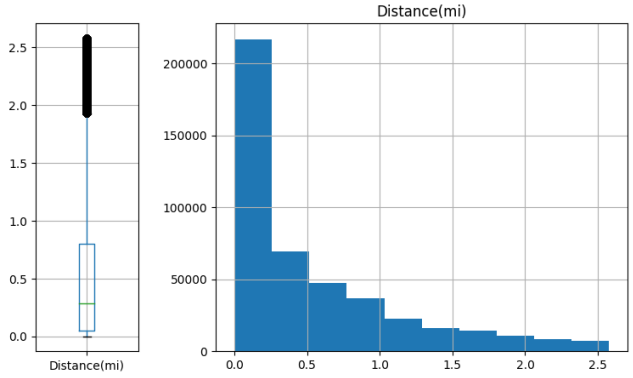

Figura 6: Boxplot i Histograma de Distància sense atípics.

3.5 *Second Feature selection*

Un altre cop hem de veure quines variables són significatives per tal d'explicar la nostra variable resposta. En aquest segon pas, no ho fem servir la lògica implementada inicialment, sinò que ho fem a partir de la matriu de correlació entre variables. La multicol·linealitat, és a dir l'alta avaluació entre variables independents, pot ser problemàtica. Per tant, si algunes variables estan molt correlacionades entre si de la matriu d'autocorrelació hauríem de considerar eliminar algunes d'aquestes. A la Figura 1 de l'Apèndix s'ha representat aquesta matriu, ens hem de fixar en les columnes que estan en blanc. Per això mateix prenem la desició de treure les variables *Visibility* i *Turning Loop*.

3.6 *Normalization*

Normalitzar les dades és una tasca crucial per l'aprenentatge automàtic ja que pot arribar a eliminar biaix. Facilita interpretacions significatives ja que els algorismes no estan influits per diferències d'escala entre característiques.

En el nostre cas, hem aplicat una estandarització a les variables numèriques *Distance(mi)*, *Temperature(F)*, *Humidity(%)*, *Pressure(in)* ja que podem veure que la seva distribució tendeix a una gaussiana. Això mateix correson a fer l'operació $\frac{X-\mu}{\sigma}$ que transforma les dades en una normal $N(\mu = 0, \sigma = 1)$. Encara que les variables *Start_Lng* i *Start_Lat* es consideren numèriques no creiem necessari normalitzar-les ja que son coordenades.

3.7 *Ending of the preprocessing*

Finalment, en aquesta etapa s'ha fet un remostreig de les dades preprocessades i s'han guardat en quatre fitxers format *.csv* on tenim *test* i *train* per a les variables i les prediccions.

4 Protocol de Validació

4.1 Mètriques de validació

És molt important establir les mètriques de validació quan obtenim resultats dels diferents tipus de models per comparar-los. De fet, quan treballem amb un *target* desequilibrat per classificació és essencial triar mètriques d'avaluació que proporcionin informació sobre el rendiment més enllà de la simple *accuracy*.

Així doncs, les mètriques que utilitzarem son *accuracy*, *precision*, *recall* i *F1 Score*. És interessant destacar que la precisió és una mètrica habitual, encara que pot arribar a ser enganyosa per a dades desequilibrades. Pel que fa referència al *recall*, *precision* i *F1 Score* hi ha de tres tipus: *macro*, *micro* i *weighted*. Aquestes depenen de l'objectiu i les característiques del conjunt de dades. En general, utilitzem *micro* si tenim en compte el rendiment general i el desequilibri de classe. D'altra banda, *macro* si tenim en compte el rendiment per a cada classe amb igualtat. Finalment, *weighted* si volem un equilibri entre el rendiment global i el rendiment específic de la classe, especialment en presència de desequilibri de classe. Per aquesta raó, la mètrica que tindrem en compte principalment per a les tres mètriques serà la *weighted* ja que les dades es generen de forma desbalancejada sistemàticament. Concretament, seleccionarem els models amb la *F1-score weighted* més alta possible ja que ens interessa un bon rendiment tant per la precisió i el *recall*. Per a valors baixos de la variable a predir ens interessa tenir més rendiment a la *precision* per penalitzar falsos positius i per a valors més alts, ens interessa tenir més rendiment al *recall* ja que d'aquesta manera penalitzem el nombre de falsos negatius.

4.2 Mètode de remostreig

Aquesta tasca també és molt important ja que ens ajudarà a obtenir millors resultats i més fiables. Per tal de fer prediccions i evaluar les mètriques de validació sobre els nostres models, hem de fraccionar les dades. Per aquest treball, hem considerat dividir fer dos grups: el 67% seran d'entrenament i validació, mentre que el 33% formaran el conjunt de test.

De la mateixa forma, hem de considerar que la majoria de models tenen hiperparàmetres. Per aquesta raó farem servir la tècnica de *k-fold cross-validation*. Utilitzarem $k = 5$, ja que és una bona manera de fraccionar les dades. Encara que la millor estratègia seria utilitzar el *loocv* no ho podem fer servir degut al gran volum de dades que hi ha i seria molt costós computacionalment.

Finalment, hem d'estimar la capacitat predictora del model. Aquesta part es farà amb el model entrenat amb les dades d'entrenament i els millors hiperparàmetres que hem obtingut anteriorment per a cada model en específic. La capacitat predictora la mesurarem amb les mètriques de validació sobre les dades d'entrenament ja que en tots els models estem fent *cross-validation*. Finalment,

escullirem els tres millors models i els evaluarem amb les dades de *test* per tal d'obtenir uns resultats finals.

La Figura 2 que trobem a l'apèndix mostra de forma il·lustrativa aquest mètode de remostreig per a $k = 5$.

5 Mètodes de Modelització

En aquesta secció s'indiquen quins models hem ajustat per predir els resultats i perquè hem seleccionat aquests en específic. És important destacar que partim d'un gran conjunt de dades, cosa que ens impideix seleccionar models molt costosos computacionalment. Per exemple, utilitzar el model *support vector machin* no seria massa intel·ligent ja que en el pitjor dels cassos tenen un cost computacional de $O(n^2d + n^3)$ on n és el nombre de dades i d el nombre de *features*. Això és inviable per aquest conjunt de dades en específic.

També hem considerat ajustar hiperparàmetres en els models al fer *cross-validation*, però no explicarem en detall els valors que hem escullit.

5.1 Arbre de decisió

Com a primera opció, hem seleccionat un arbre de decisió per a classificació per al conjunt de dades desequilibrat ja que és molt versàtil en manejar grans conjunts de dades de manera eficient ja que és un dels menys costosos computacionalment. Els arbres de decisió són intrínsecament interpretables, la qual cosa permet una visualització i comprensió clara de la importància de les característiques i els camins de decisió. De la mateixa manera, en aquest model i tots els models de la família de l'arbre de decisió presenten aquesta versatilitat qeu podria ser de gran utilitat si es volgués traslladar en una aplicació en temps real d'un sistema que potser ha quedat una mica antic ja que és un conjunt de condicionals. Finalment, és interessant destacar que els arbres de decisió serveixen com a base per a mètodes de més complexos que es mencionaran en endevant.

5.2 *Random Forest*

El següent model de la família dels arbres de decisió que hem considerat és el *Random Forest*. Considerem que aquest model pot arribar a tenir més eficàcia ja que fa la mitjana de múltiples arbres de decisió, reduint l'*overfitting* i millorant la generalització. Són molt adequats per a dades desequilibrades ja que poden incorporar tècniques com la ponderació de classes i el submostreig equilibrat. A més, ofereixen temps d'entrenament i prediccions ràpides, encara que triguen una mica més que arbres de decisió simples.

5.3 *Extra Trees*

Aquest model l'hem triat, ja que és una variant dels *Random Forests* la qual proporciona una major diversitat en els arbres generats, això millora l'exactitud i la robustesa del model. A més a més, és molt eficient al tenir un gran volum de dades. És molt eficient computacionalment perquè no requereix d'un gran ajust de paràmetres, per tant no es consumeix molt de temps d'execució.

5.4 *Voting Classifier*

Fer un *voting classifier* amb un conjunt de models és bona idea perquè combina la interpretabilitat i rapidesa dels models i en el cas que un model en concret falli, té d'altres que potser no ho estan fent en aquell cas en concret. En el nostre cas, hem considerat fer *voting* entre l'arbre de decisió,

el *Random Forest* i el *extra trees* ja que com es veurà més endevant, son els que donen millors resultats. A més, hem considerat el *voting soft* on escullim una ponderació de probabililtats per igual de cada classe ja que d'aquesta manera podem obtenir més mètriques.

5.5 *Gradient Boosting*

El mètode del *gradient boosting* construeix el model final mitjançant petites pases, les quals són models els quals redueixen l'error produït pels models anteriors. Aquesta tècnica ens permet construir un model final amb una alta precisió en les prediccions, sobretot en conjunt de dades com aquest, el qual és gran i està desbalancejat. En comparació amb els models anteriors, aquest té un cost computacional major, i es poden ajustar hiperparàmetres per millorar el model. L'hem tingut en compte, però al final no l'hem dut a terme, ja que com s'ha dit el seu cost computacional és molt gran i per trobar els millors paràmetres, el temps d'execució de l'algoritme és molt gran.

5.6 Regressió Logística

La regressió logística és adequada per classificar variables amb múltiples objectius quan la relació entre predictors i objectius és lineal. És interpretable, computacionalment eficient i gestiona bé la multicolinearitat. Amb quatre objectius, la regressió logística proporciona un enfocament senzill però eficaç per a les tasques de classificació sense requerir recursos computacionals extensos o ajustaments complexos.

5.7 *Quadratic Discriminant Analysis*

Hem decidit procedir amb un mètode no lineal: QDA. La raó ha sigut que les matrius de covariància de totes les classes no són iguals en teoria. A més a més, aquest mètode és útil quan les dades d'entrada segueixen una distribució Normal, i aquest és el cas perquè hem escalat les variables a l'apartat del preprocessament.

5.8 *Linear Discriminant Analysis*

De la mateixa manera que el QDA, hem seleccionat aquest model. El LDA suposa que cada classe té la mateixa matriu de covariàncies. Al preprocessament ja vam veure com segurament això no era cert, però anem a ajustar aquest model ja que la resposta no era del tot clara i potser ens dona millors resultats que el QDA. Cal tenir en compte que en cas que sigui millor, aquest mètode és lineal i per tant molt menys costós computacionalment.

5.9 *Naive Bayes*

Aquest mètode el podem aplicar si les dades són independents entre elles. Això no està clar, ja que hem vist a la Figura 1 de l'apèndix com la correlació entre variables és molt petita però hi és. Per tant en teoria aquest mètode no hauria de donar bons resultats, però al ser tant petites les correlacions potser podem aplicar aquest mètode. De la mateixa manera, hem considerat fer un *Categorical* NB i un *Numerical* NB i combinar els resultats. Compararem aquesta combinació amb el *Gaussian Navie Bayes* original.

5.10 k-*Nearest Nieghbours*

Per últim, val la pena seleccionar el mètode de k-Nearest Neighbors (k-NN) perquè és un model senzill i intuïtiu que sobresurt en el maneig de dades desequilibrades, especialment amb una elecció acurada de la mètrica de distància i el valor k. Hem de ser molt curossos a l'hora de seleccionar

aquest hiperparàmetre k ja que per valos baixos podem introduir *overfitting* i per valos alts *underfitting* cosa que ens indica que ja ens serveix fer validació creuada en aquest model. Tot i ser computacionalment intensiu, la seva eficàcia per proporcionar prediccions molt precises pot superar el temps de processament més llarg en comparació amb models més ràpids com els arbres de decisió.

5.11 *Clustering*

Per últim hem considerat una tècnica d'aprenentatge no supervisat. El cas és que en el nostre dataset les classes ja estan definides, és per això que el que farem serà aplicar *K-Means* i *EM* amb 4 clústers. I per poder fer les comparacions amb la resta de mètodes supervisats, veurem si els punts que haurian d'anar a una certa classe s'han classificat correctament o no amb les mateixes mètriques que els altres. Si no tinguèssim dades etiquetades, no podríem treure aquestes mètriques amb *clustering*, però com tenim un *ground truth* ho podem fer servir. De la mateixa manera, es podria fer servir el *CH*, *SS* o el *DB*, però per fer comparacions entre models no creiem que representin del tot bé la diferència entre aquests.

6 Resultats de la modelització

Una vegada hem ajustat tots els models anteriors amb *cross-validation* es mostren els resultats de validació. Aquests resultats es poden veure a partir del codi del *notebook model.ipynb*. Aquests corresponen als de la Taula 1.

Model	Accuracy	Recall	Precision	F1 Score
Random Forest	0.839107	0.839107	0.819237	0.814196
Voting Classifier	0.836736	0.836736	0.815269	0.813440
Decision Tree	0.819103	0.819103	0.797032	0.804470
Extra Trees	0.824266	0.824266	0.798212	0.793315
QDA	0.772377	0.772377	0.779837	0.773086
k-Nearest Neighbor	0.786571	0.786571	0.758315	0.768350
GNB	0.765089	0.765089	0.773222	0.765268
LDA	0.800461	0.800461	0.751742	0.760201
GNB numerical	0.764451	0.764451	0.741136	0.748999
GNB combined	0.800026	0.800026	0.643670	0.711274
GNB categorical	0.800055	0.800055	0.640088	0.711187
Logistic Regression	0.800050	0.800050	0.640087	0.711185
EM	0.581016	0.581016	0.581016	0.581016
k-Means	0.368645	0.368645	0.368645	0.368645

Taula 1: Taula comparativa dels resultats dels models **evaluats en el conjunt de dades de *train* fent *cross-validation*** fent $K = 5$.

6.1 Sel·lecció de models

Per fer la selecció de models ens basarem en la Taula 1. Fins ara, el model amb el que tenim millors resultats correspon al model del *Random Forest*. Això és el que diuen els resultats, però com veiem a la Taula 1, la diferència de l'F1 Score entre l'arbre de decisió o el *Voting Classifier* i el *Random Forest* és petita. Encara que la resta de mètriques així com recall o precision surt guanyant el *Random Forest*. També cal dir que el *Random Forest* és una combinació d'arbres independents,

per tant, la seva variància és més petita que la d'un sol arbre de decisió. Seguidament, trobem el model *Extra Trees* que hem utilitzat per fer prediccions amb el *Voting Classifier*.

També s'ha de mencionar que altres models que no pertanyen als de la família de l'arbre donen bons resultats. En són exemples, el k-NN, QDA i LDA. El k-NN té una precisió notable però la seva F1 Score és inferior comparat amb altres models. El QDA, tot i que té una precisió i un F1 Score inferiors als millors models, és una opció interessant per la seva simplicitat i velocitat de càlcul. La LDA, encara que té resultats inferiors al QDA, mostra una tendència similar i pot ser útil en casos on la distribució de les dades sigui més propera a la normalitat.

Pel que fa a la regressió logística, dona un rendiment notable en l'F1 Score, similar al dels models basats en arbres. Això pot ser degut a la seva capacitat per gestionar relacions lineals entre les característiques.

Finalment, és interessant destacar el baix rendiment de la classificació utilitzant mètodes d'aprenentatge no supervisats. Concretament, el rendiment de l'algoritme k-means és baix. No obstant això, si utilitzem l'algoritme de *clustering Gaussian Mixture* (EM), tenim un millor rendiment. Això pot passar degut a la distribució de les dades ja que, tal com havíem vist en el preprocessament, moltes variables seguien una distribució normal o gaussiana.

En resum, tot i que el *Random Forest* té els millors resultats en general, la diferència no és tan gran en comparació amb altres models com el *Voting Classifier* i el *decision tree*. La selecció final d'un únic model podria dependre també d'altres factors com la complexitat del model, el temps de càlcul, i les característiques específiques del problema a resoldre. Com nosaltres volem escullir tres models, hem escullit aquests tres models en concret.

7 Validació final

Una vegada hem seleccionat els tres millors models, anem a provar el rendiment final per a la tasca de classificació amb el conjunt de dades de *test*.

En primer lloc, considerem treure les mateixes mètriques que abans però ara tenint en compte les dades de *test*, per comprovar que no hi hagi *overfitting* ja que els models que pertanyen a la família dels arbres és força comú que hi hagi *overfitting*.

Model	Accuracy	Recall	Precision	F1 Score
Voting Classifier	0.837592	0.837592	0.816713	0.811317
Decision Tree	0.822398	0.822398	0.800692	0.807842
Random Forest	0.836449	0.836449	0.817707	0.805207

Taula 2: Taula comparativa dels resultats dels models **evaluats en el conjunt de dades de *test***.

Tal i com podem veure a la Taula 2, les mètriques de F1 Score son bastants similars per als tres models en el conjunt de *test*. Pel que fa referència a valors òptims, el model *voting classifier* és el que dona millors resultats, seguit de l'arbre de decisió i el *Random Forest*.

A continuació, anem a mostrar el gràfic ROC per als tres tipus de model. El gràfic ROC, *Receiver Operating Characteristic*, és crucial per comparar models perquè mostra la capacitat d'un model per distingir entre classes. Permet avaluar el rendiment del model en diferents llindars de classificació. L'àrea sota la corba ROC rep el nom d'AUC i si és més gran indica un millor rendiment. Comparar els AUC de diversos models facilita identificar quin és més eficaç en discriminar entre classes. Aquests gràfics serà una altre eina per comparar els models.

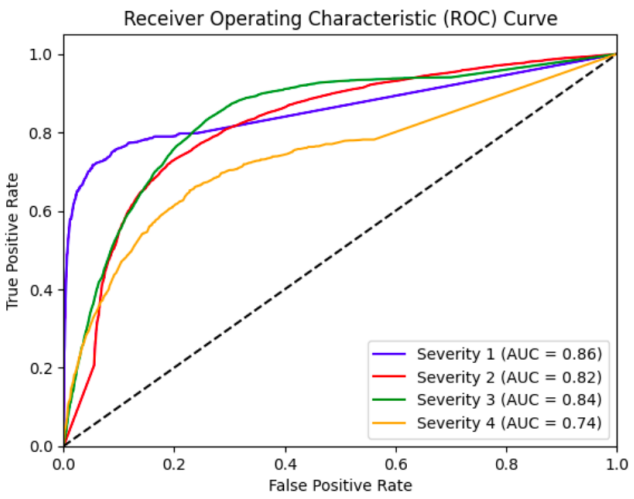

Figura 7: Gràfic ROC per a la classificació dels quatre tipus de la variable *target* per a l'arbre de decisió.

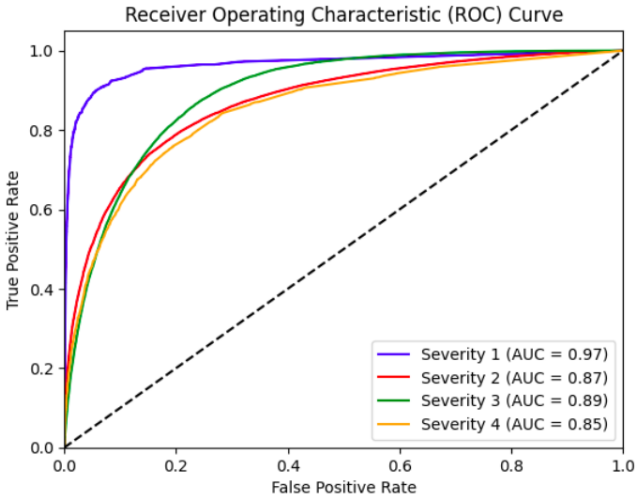

Figura 8: Gràfic ROC per a la classificació dels quatre tipus de la variable *target* per al *Random Forest*.

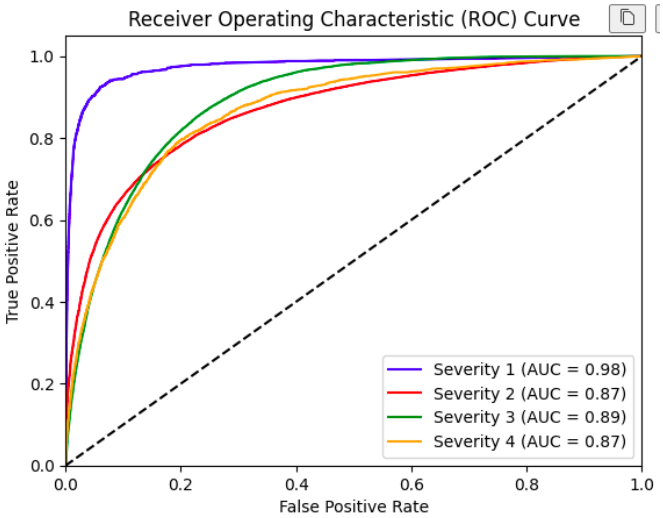

Figura 9: Gràfic ROC per a la classificació dels quatre tipus de la variable *target* per al *voting classifier*.

Si ens fixem a les figures 7, 8 i 9, podem veure com clarament el model de l'arbre de decisió és el que obté pitjors resultats ja que la curva tendeix a ser més irregular. De fet, la curva per als models del *Random Forest* i el *voting* son bastant semblants, cosa que ens indica que el *voting* està prioritzant més el *Random Forest* que l'arbre de decisió. Això té sentit ja que tal i com havíem vist per l'entrenament amb *cross-validation*, les millors mètriques les obteníem amb el *Random Forest*. Per últim, és interessant destacar que el gràfic de la Figura 9, corresponent al *voting* és el que obté millors resultats.

Un altra gràfic que seria interessant analitzar és el de la importància dels *features*. Aquest gràfic mostra la importància dels *features* de cara a la predicció d'un model. Ho farem en els següents dos models: tant l'arbre de decisió com el *Random Forest*. Encara que no ho podem fer en el *voting* ja que és una combinació de models, el resultat serà molt similar al del *Random Forest*, tal i com havíem vist també en el cas del ROC.

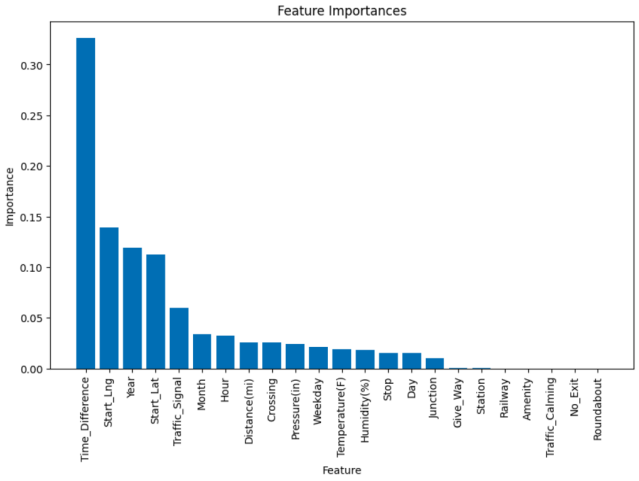

Figura 10: Gràfic que mostra la importància de les característiques en el model de l'arbre de decisió.

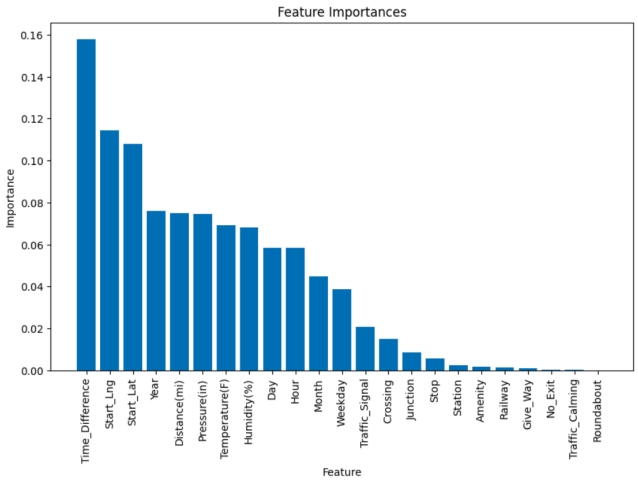

Figura 11: Gràfic que mostra la importància de les característiques en el model del *Random Forest*.

Tal i com podem observar a ambdos gràfics corresponents a la Figura 10 i 11, la variable creada quan hem fet el *feature engineering Time_Difference* és la més significativa de cara a fer prediccions. No obstant això, hi ha una gran diferència entre el percentatge de importància per als dos models diferents. Al gràfic corresponent a l'arbre de decisió les probabilitats no estan tant uniformament distribuides que al del *Random Forest*. Encara que no passa en aquest tal i com hem vist a la Taula 2, podria arribar a indicar *overfitting* ja que només està prioritzant una característica.

Finalment, una vegada hem fet una validació final sobre els dos models triats, seria interessant fer un anàlisi més profund sobre les mètriques de cada categoria a la classe a predir. Degut a que els dos models triats treuen mètriques semblants, només es farà pel *voting classifier* ja que ha sigut el que ha tret millors resultats per aquest conjunt de *test*. Això mateix es pot veure reflexat en la Taula 3.

Tal i com s'ha indicat abans, estaria bé obtenir un bon resultat en la presició per a valors 1 i 2 de la gravetat ja que això ens indica que no hi ha molts accidents que es detecten greus però en realitat ho son. D'altra banda, també seria interessant obtenir bon resultat al *recall* per al valors 3 i 4 ja que ens indicaria que no hi ha molts accidents que son falsos positius. És a dir, que no es detecten com a accidents greus però en realitat sí que ho son.

Classe	Recall	Precision	F1 Score
Severity 1	0.26	0.70	0.38
Severity 2	0.96	0.85	0.91
Severity 3	0.36	0.68	0.47
Severity 4	0.03	0.52	0.07

Taula 3: Taula comparativa dels resultats de cada classe de la variable a predir.

Tal i com podem observar a la Taula 3, la precisió per a accidents no greus és prou bona ja que està per sobre del 70% per a *Severity 1* i per a *Severity 2* tenim un molt bon rendiment amb un 85%. De la mateixa manera, és interessant destacar el molt bon rendiment per a la classe 2 per a totes les mètriques.

No obstant això, encara que la precisió per a accidents greus sí que és prou bona, la mètrica important que és el *recall* no ho es pas. Fent referència al *recall*, per al cas *Severity 3*, el model ho fa una mica millor que un classificador aleatori ja que el rendiment d'un classificador aleatori en

aquest cas és un 25%, però el rendiment no és molt bo ja que és d'un 36%. El que sí que es crític és el rendiment per a *Severity 4* ja que és d'un 0.03. Això és molt greu, ja que recordem que el *recall* penalitza falsos negatius. El nostre model tendeix a predir molts accidents greus com a no greus.

7.1 Cas real

Com a exercici final, un dels autors d'aquest projecte va cap a Estats Units aquest estiu. Concretament, estarà per la zona oest. Déu no vulgui que no passi cap desgràcia, però anem a calcular com de greu seria un accident fent servir tot el que hem anat fent durant el projecte en cas de tenir-ne un. En primer lloc, podriem arribar a considerar que el model predirà que l'accident serà de grau 2 tal basant-nos en la Taula 3.

Seguint amb el parràgraf anterior, hem creat un altre arxiu per crear un nou *DataFrame* amb unes dades que suposarem. Aquestes dades es poden veure al *notebook predict_autor.ipynb*. Si és d'interès mirar els valors de les dades que hem suposat, podeu consultar l'arxiu mencionat. Tal i com havíem mencionat anteriorment, el model *Random Forest* prediu el cas de *Severity 2*. Si recordem el significat d'aquest valor era un accident amb lesions lleus, bon indicador.

Classe	Probability
Severity 1	0.00452808
Severity 2	0.83517324
Severity 3	0.16029869
Severity 4	0

Taula 4: Taula comparativa de la probabilitat per al cas real de la variable a predir.

8 Conclusions

Després d'analitzar els resultats obtinguts dels diferents models utilitzats per predir la gravetat dels accidents de trànsit, podem extreure diverses conclusions importants.

Pel que fa al rendiment dels models, els models de *Random Forest* i *Voting* presenten resultats similars, indicant que el model de *Voting* prioritza el Random Forest sobre l'arbre de decisió, cosa que és consistent amb les mètriques d'entrenament inicials que van mostrar que *Random Forest* oferia les millors mètriques.

Quant a la importància de les característiques, la variable creada durant el *feature engineering* denominada *Time_Difference* va resultar ser la més significativa per fer prediccions. Tanmateix, hi ha una diferència notable en el percentatge d'importància entre els models d'arbre de decisió i *Random Forest*, amb distribucions menys uniformes en l'arbre de decisió, cosa que podria indicar sobreajustament.

En l'anàlisi de mètriques, la precisió per als accidents no greus (Severity 1 i 2) és força bona, amb una precisió superior al 70% per a Severity 1 i un rendiment excel·lent per a Severity 2 amb un 85%. En contrast, encara que la precisió per als accidents greus és adequada, el recall és una àrea crítica. En el cas de Severity 3, el model millora marginalment sobre un classificador aleatori amb un 36%, però per a Severity 4, el rendiment és extremadament baix amb un 3%, cosa que indica que el model tendeix a classificar incorrectament molts accidents greus com a no greus.

En una simulació d'un cas real, el model va predir correctament un accident de gravetat 2, cosa que és indicativa de lesions lleus, demostrant un bon rendiment pràctic i esperat del model en

escenaris reals.

En resum, encara que s'ha assolit un bon rendiment en la classificació d'accidents menys greus, és necessari millorar el *recall* per als accidents greus per reduir el nombre de falsos negatius. Això és crucial per a aplicacions pràctiques on la identificació precisa d'accidents greus pot tenir un impacte significatiu en la resposta d'emergència i en la seguretat viària en general. No obstant això, per fer-ho necessitaríem mes dades sobre accidents greus per no tenir tant desbalanceig en les dades.

Per a futurs treballs, es recomana explorar tècniques addicionals de balanceig de dades i models més sofisticats que puguin manejar millor els casos d'accidents greus per millorar el *recall* en aquestes categories.