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

Aprenentatge Automàtic 1

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

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

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* 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*

[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

- 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 balencejar. No obstant això, a primera vista podem pensar que balencejar 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 com son 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.

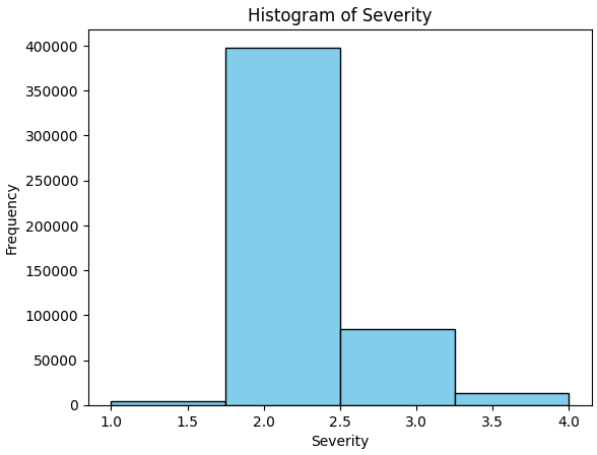

Figura 1: Histograma de la variable *Severity*.

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, o *Wind_Chill(F)*. 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. Ens estaria donant 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.

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.

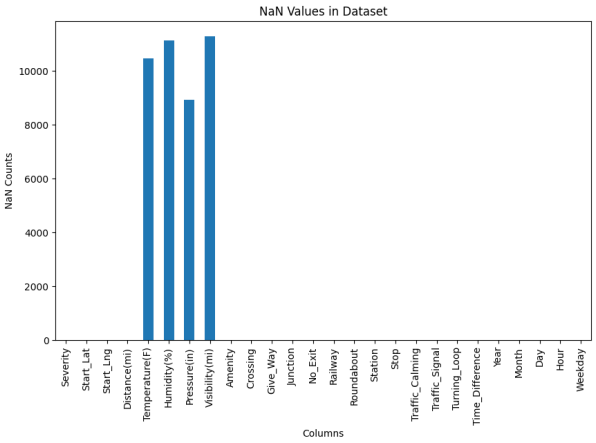

Figura 3: Diagrama de barres dels Nan's abans inspeccionar Nan's.

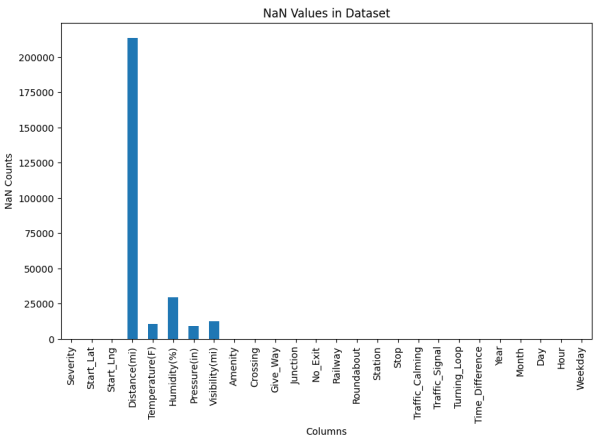

Figura 4: Diagrama de barres dels Nan's després inspeccionar Nan's.

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.

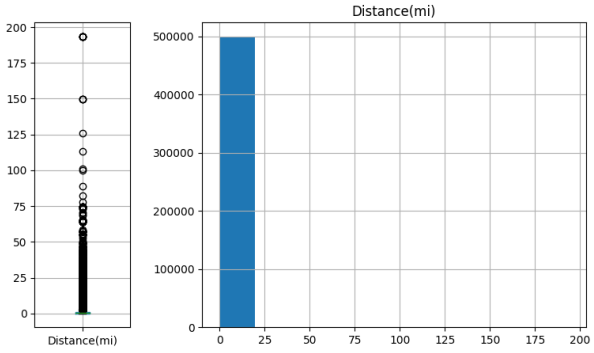

Figura 5: Boxplot i Histograma de Distància amb atípics.

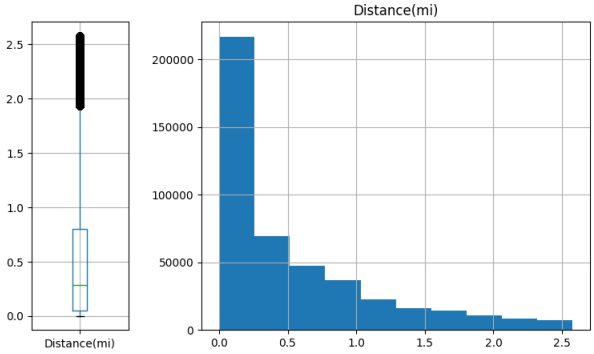

Figura 6: Boxplot i Histograma de Distància sense atípics.

3.5 *Second Feature selection*

Un altre cop hem de veure quines variables són significatives per tal d'explicar la nostra variable resposta. Ara no farem servir la lògica implementada inicialment, sinò que ho farem a partir de la matriu de correlació entre variables. A la Figura 8 de l'Apèndix s'ha representat aquesta matriu, ens hem de fixar en les columnes que estan en blanc ja que no tenen res a veure amb la resta de variables. Per això mateix prenem la desició de treure les variables *Visibility* i *Turning Loop* ja que no estan correlada amb cap variable més.

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 un fitxer format *.csv* amb nom *clean_data.csv* que servira per ajustar els models.

4 Protocol de Validació

4.1 Mètriques de validació

És molt important establir en primer lloc les mètriques de validació quan obtenim els 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 precisió.

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 a l'*accuracy*, *precision* i *F1 Score*, hem de triar entre *macro*, *micro* i *weighted* depèn 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 serà tant el *recall weighted*, el *precision weighted* i el *F1 score weighted*.

4.2 Mètode de remostreig

Aquesta tasca 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 fer que el model entreni amb unes dades i mirar la seva capacitat predictora amb les altres dades. Dividirem les dades en dos grups, el 67% seran d'entrenament i validació, mentre que el 33% del conjunt de test.

Per altra banda, els models tenen hiperparàmetres, paràmetres els quals hem de fixar abans d'entrenar el model i que són necessaris. Farem servir la tècnica de *k-fold cross-validation*, ja que ens sembla la millor per estimar aquests paràmetres. Utilitzarem $k = 5$, ja que ens sembla una bona manera de fraccionar les dades i el propi mètode de python *GridSearchCV* ens donará el o els paràmetres que millor ajusten les dades, tot basant-nos en l'*F1 Score Weighted*, ja que les dades no són balancejades. Encara que la millor estratègia seria utilitzar el *loocv*, no ho podem fer servir degut a la naturalesa de les dades.

Finalment, hem d'estimar la capacitat predictora del model. Aquesta part es fará amb el model entrenat amb les dades d'entrenament i els millors hiperparàmetres que hem obtingut anteriorment. La capacitat predictora la mesurarem amb les mètriques de validació sobre les dades de test.

La Figura 7 mostra de forma il·lustrativa aquest mètode de remostreig.

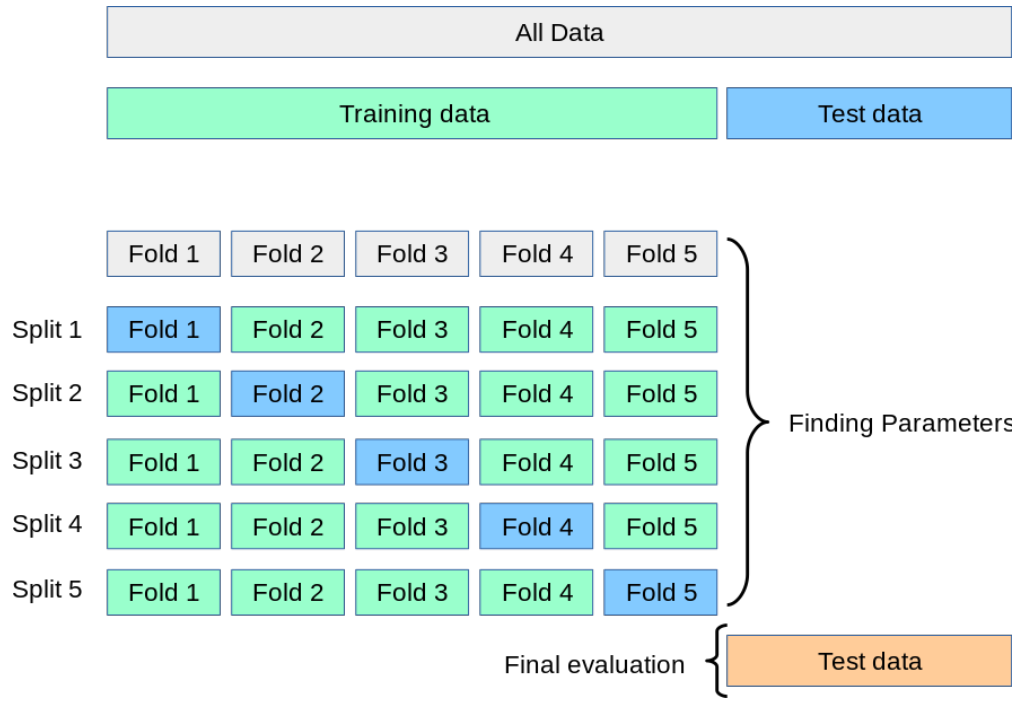

Figura 7: Esquema on es mostra la tècnica de *cross-validation* utilitzada per aquest problema. Tal i com s'indica a la imatge, s'utilitza *cross-validaton* a les dades d'entrenament per trobar les hiper-paràmetres i després s'evalua amb les dades del *test.*

5 Mètodes de Modelització

En aquesta secció s'indiquen quins models hem ajustat per predir els resultats. És interessant destacar que hem utilitzat tant mètodes lineals com no lineals.

5.1 Arbre de decisió

El primer model que hem evaluat ha sigut el *DecisionTreeClassifier*. Aquest mètode és ideal per classificar a causa de la seva simplicitat, interpretabilitat i capacitat de manejar dades tant numèriques com categòriques. Gestiona automàticament les interaccions de característiques i les relacions no lineals, la qual cosa la fa eficient per a conjunts de dades de mida petita i mitjana sense un preprocessament extens. També es interessant destacar que és un dels mètodes que triga menys temps a l'hora d'ajustar els paràmetres.

5.2 Logistic Regression

En segon lloc, 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.

5.3 Quadratic Discriminant Analysis

En tercera instància 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 per tant no podem aplicar LDA, i a més a més, és un bon mètode per classificar donat que les variables d'entrada segueixen una Gaussiana.

6 Resultats de la modelització

Una vegada hem ajustat tots els models anteriors. Els resultats, igual que anteriorment es poden observar a la Taula 1.

Model	Accuracy	Recall	Precision	F1 Score
DecisionTree	0.810473	0.810473	0.797908	0.803417
Logistic Regression	0.798688	0.798688	0.637903	0.709298
QDA	0.769952	0.769952	0.777368	0.770625

Taula 1: Taula comparativa dels resultats dels models **evaluats en el conjunt de dades de *test***.

Fins ara, el model amb el que tenim millors resultats correspon al model de l'arbre de decisió per classificació.

7 Apèndix

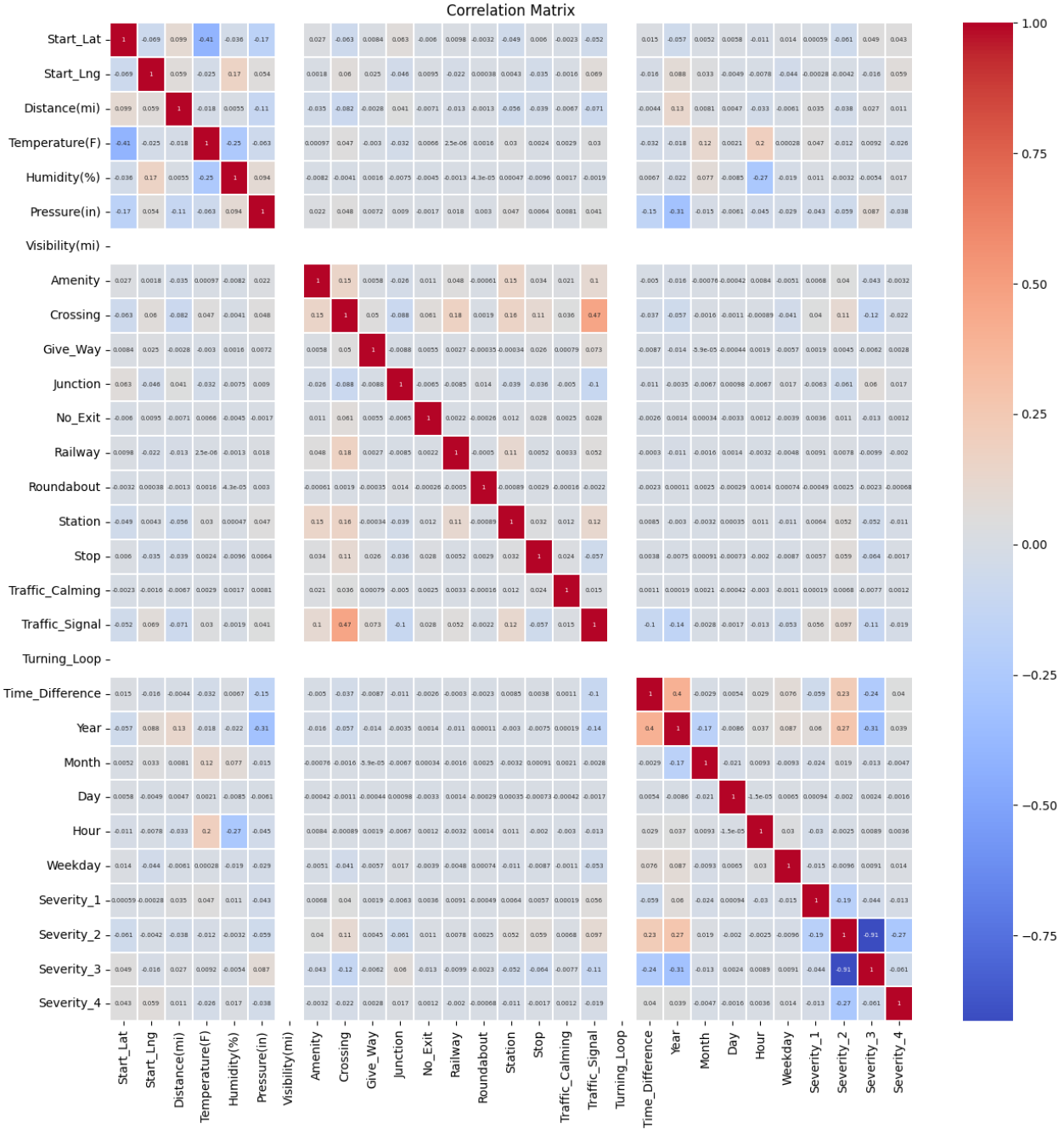

Figura 8: Matriu de correlació entre les variables del per a dur a terme