Classificació de Supervivents del Titanic

Ricard Urpí Vilanova[a,1], **Ferran Villarta**[b,2] **and Natan Sisoev**[c,3]

[a] *1711326*
[b] *1704051*
[c] *1706198*

Abstract—Aquest treball presenta una anàlisi exhaustiva per a la classificació de supervivents del Titanic mitjançant tècniques d'aprenentatge automàtic. A partir d'un dataset amb informació sobre passatgers (classe, edat, sexe, tarifa, etc.), es realitza un preprocessament detallat que inclou imputació de valors perduts amb KNN, extracció de característiques com el deck de la cabina i títols dels noms, i normalització de dades. Es comparen 12 models diferents utilitzant Repeated Stratified K-Fold Cross Validation amb F1-score com a mètrica principal. El Gradient Boosting amb paràmetres optimitzats (learning rate 0.1, max depth 3, 201 estimators) obté els millors resultats amb un F1-score de 0.75 i AUC-ROC de 0.85, utilitzant només tres característiques: classe de passatger, tarifa i sexe. El model demostra un bon equilibri entre precisió i recall, amb capacitat de generalització adequada i sense sobreajust significatiu.

Keywords—*Titanic, Aprenentatge automàtic, Gradient Boosting, Classificació, Cross Validation, F1-score, ROC-AUC*

Contents

1. EDA (exploratory data analysis)

Llegint i analitzant la base de dades vam veure que teniem les següents variables:

- **PassengerId**: etiqueta, identifica a cada persona
- **Survived**: binària, 1 = sí, 0 = no
- **Pclass**: categòrica, 1 = primera, 2 = segona, 3 = tercera classe
- **Name**: etiqueta, nom de la persona (potser no únic)
- **Sex**: categòrica, female o male
- **Age**: numèrica, edat en anys
- **SibSp**: numèrica, nombre de germans o parelles en el Titanic
- **Parch**: numèrica, nombre de pares o fills en el Titanic
- **Ticket**: text, codi del tiquet
- **Fare**: numèrica, preu del tiquet
- **Cabin**: text, codi de la cabina
- **Embarked**: categòrica, port d'embarcació: C = Cherbourg, Q = Queenstown, S = Southampton

El target és l'atribut **Survived**, que pot prendre valors 0 o 1.

Ens faltaven dades, el port d'embarcació per a 2 persones, l'edat per a 177 i la cabina per a 687.

Analitzant les correlacions entre variables i la distribució de les dades no trobem grans inconvenients i ens cal acabar de seleccionar, normalitzar i processar les dades.

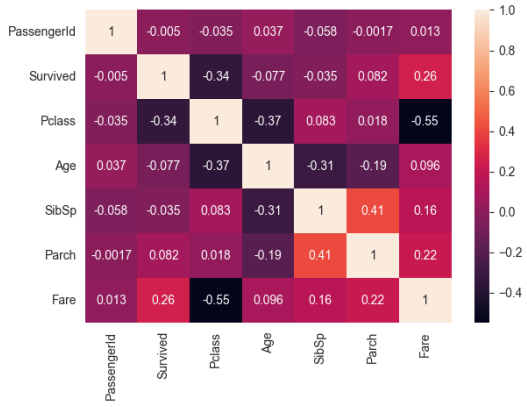

Figure 1. Matriu de correlacions

2. Preprocessing

Per tractar les dades que ens falten hem començat per la cabina, considerem que és molt important per a la predicció, ho podem veure en la següent fotografia.

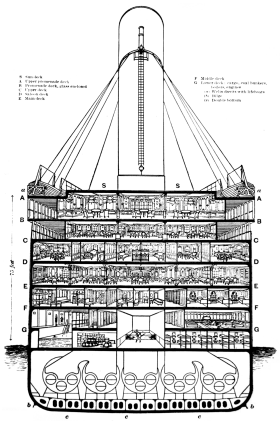

Figure 2. Decks

Com ens falten moltes dades (aproximadament 80% dels passatgers no tenen cabina) no té sentit assignar una cabina a cada passatger, hem creat una nova variable que conté la lletra del codi de la cabina i als passatgers que no en tenen els hem assignat una *X*. Afegim una variable que indica si tenen cabina o no. Als passatgers que no tenien port hem considerat que podíem mirar passatgers que tinguin preus de tiquets similars i assignar a aquests passatgers aquest port. En el cas de l'edat hem decidit imputar les dades que ens falten (20% aproximadament). La mitjana d'edat varia molt entre classes: les classes més pobres tenen una mitjana d'edat menor, així que hem creat una variable que ens indica el nombre de familiars (més familiars pot indicar una edat més baixa). A més, tots els noms tenen un títol i aquests corresponen majoritàriament a una certa franja d'edat, sexe... Hem creat una variable amb el títol de cada persona. Aquest és un cas ideal per usar KNN i trobar passatgers similars per assignar una edat als que no tenen dades. Per això, considerem que les variables rellevants són **Fare**, **SibSp**, **Parch**, **FamilySize**, **Title**, **Cabin_missing** i **Pclass**. Afegim variables binàries per indicar el deck (coberta

que correspon a la lletra del codi del tiquet), el port d'embarcament i el sexe. Finalment, tractem amb els *outliers* i normalitzem les dades. El nostre dataset ens queda amb les següents variables ja normalitzades: **Sex_male**, **Embarked_Q**, **Embarked_S**, **Deck_B**, **Deck_C**, **Deck_D**, **Deck_E**, **Deck_F**, **Deck_G**, **Deck_X**, **Title_Military**, **Title_Mr**, **Title_Mrs**, **Title_Ms**, **Title_Nobility**, **Title_Rev**, **Age**, **SibSp**, **Parch**, **Fare**, **FamilySize**, **Pclass**, **PassengerId**, **Survived** i **Cabin_missing**.

3. Metric selection

En aquest apartat hem seleccionat la mètrica que millor s'ajusta al nostre model.

Després d'entrenar el model utilitzant regressió logística, obtenim les següents conclusions:

- **1. Accuracy**: Pot ser enganyosa en datasets desequilibrats. Encara que no és extremadament desequilibrat, hi ha més morts que supervivents. Així que un model que prediu gairebé sempre "mor" tindria bona accuracy, però no seria bo per detectar supervivents.

- **2. Precision i Recall**: Precision mira quants supervivents predits realment sobreviuen (minimitza falsos positius).
  Recall mira quants supervivents reals hem detectat (minimitza falsos negatius).
  Per comparar models cal un equilibri entre ambdues.

- **3. Average Precision Score**: Resumeix tota la Precision-Recall curve. És útil quan el model retorna probabilitats i hi ha molts casos positius escassos. En aquest cas les classes no són extremadament desequilibrades.

- **4. ROC-AUC**: També resumeix la capacitat del model per separar classes. Però dona menys informació en casos desequilibrats, pot donar una falsa sensació de bon rendiment quan hi ha molts negatius.

- **5. F1-score**: Combina Precision i Recall en una mitjana, equilibrant la capacitat de detectar supervivents (recall) i no predir falsament supervivents (precision).

Per aquests motius, hem triat l'F1-score com a mètrica principal per avaluar els models.

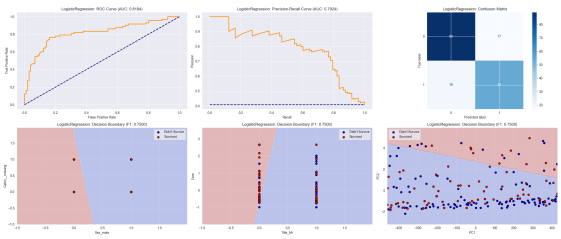

Figure 3. Gràfics

Podem observar com hem generat diferents gràfics representant les dues corbes, la matriu de confusió, les fronteres per dues parelles de features i PCA.

4. Model Selection amb Crossvalidation

Els mètodes que hem provat per fer validació creuada han estat: K-Fold, Stratified K-Fold, Repeated Stratified K-Fold i Shuffle & Split. Els hem provat tots amb cadascun dels mètodes que presentarem a continuació. Com és un dataset petit, hem decidit escollir el més robust tot i tardar una mica més: Repeated Stratified K-Fold. L'implementem directament en la nostra classe `Metrics`, de manera que qualsevol testeig que fem serà fent servir aquest tipus de validació creuada.

Seleccionem els següents models:

- **Logistic Regression**
- **KNN**
- **Gradient Descent**
- **Random Forest**
- **Extra Trees**
- **Support Vector Machine**
- **LinearSVC**
- **Gradient Boosting**
- **Naive Bayes**
- **Perceptron**
- **Passive Agressive**
- **Neural Network**

Els més adequats són Random Forest, Extra Trees i Gradient Boosting, que gestionen bé petites mostres i desequilibris. Models lineals com Logistic Regression i SVC poden funcionar si les dades estan normalitzades, mentre que Perceptron i Passive Aggressive són menys útils. Altres com KNN, Naive Bayes o MLPClassifier poden donar resultats raonables però requereixen cura amb els hiperparàmetres o tenen risc d'*overfit*.

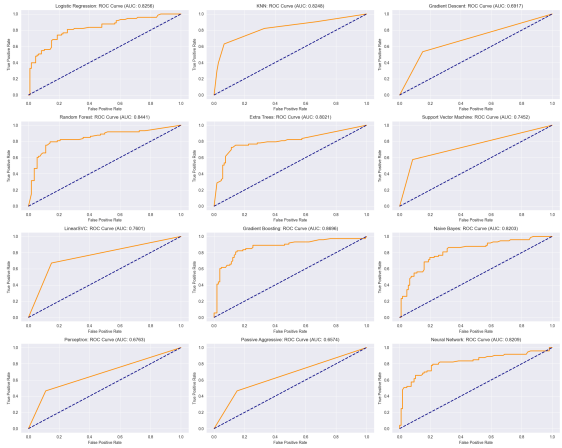

Figure 4. Corbes amb el dataset reduït

Hem provat a reduir la dimensionalitat del model per veure si obteníem bons resultats en dimensions menors que poguéssim visualitzar. Aquests és un dels molts resultats visuals que hem obtingut:

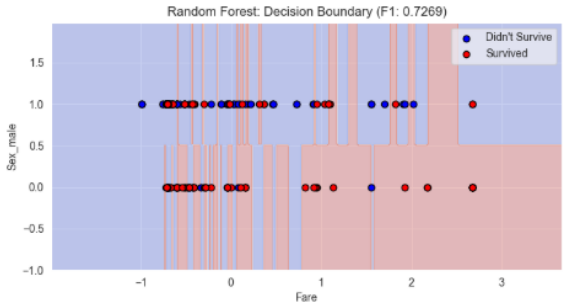

Figure 5. Random Forest amb variables Fare i Sex_male

Model	Mean	Best Params
Logistic Reg.	0.7106	C=1, max_iter=1000, penalty=l2
KNN	0.7098	metric=euclidean, n_neighbors=9
Grad. Descent	0.7109	alpha=0.001, eta0=0.01, lr=optimal
Random Forest	0.7042	max_depth=5, n_estimators=101
Extra Trees	0.7101	max_depth=2, n_estimators=51
SVM	0.7101	C=0.1, gamma=scale, kernel=linear
LinearSVC	0.7101	C=0.1, loss=hinge
Grad. Boost.	**0.7265**	lr=0.1, max_depth=3, n_estimators=201
Naive Bayes	0.7164	var_smoothing=1e-12
Perceptron	0.6108	alpha=1e-05, eta0=0.1
Passive Agg.	0.7101	C=0.001, max_iter=1000
Neural Net	0.7093	act=tanh, alpha=1e-05, lr=const.

Table 1. Comparació de models i paràmetres òptims.

Per trobar els millors hiperparàmetres hem fet una cerca provant-ne diferents combinacions, ja que el dataset és relativament petit. Hem pogut obtenir la següent taula:

Model	Mean Test Score	Best Parameters
Logistic Reg.	0.7106	C=1, max_iter=1000, penalty=l2
KNN	0.7098	metric=euclidean, n_neighbors=9, weights=uniform
Grad. Descent	0.7109	alpha=0.001, eta0=0.01, lr=optimal
Random Forest	0.7042	max_depth=5, n_estimators=101
Extra Trees	0.7101	max_depth=2, n_estimators=51
SVM	0.7101	C=0.1, gamma=scale, kernel=linear
LinearSVC	0.7101	C=0.1, loss=hinge, max_iter=1000
Grad. Boost.	**0.7265**	lr=0.1, max_depth=3, n_estimators=201
Naive Bayes	0.7164	var_smoothing=1e-12
Perceptron	0.6108	alpha=1e-05, eta0=0.1, max_iter=1000
Passive Agg.	0.7101	C=0.001, max_iter=1000
Neural Net	0.7093	act=tanh, alpha=1e-05, lr=adaptive, max_iter=1000

Table 2. Comparació de models amb els millors hiperparàmetres i el seu *mean test score*.

5. Anàlisi Final

Resultats finals del model: S'han explorat totes les combinacions possibles de *features* per determinar si era possible millorar el rendiment eliminant informació redundant. Els resultats indiquen que l'*score* augmenta de **0.726538** a **0.749594** quan s'utilitzen només les variables *[Pclass, Fare, Sex_male]*. Aquest conjunt de tres característiques proporciona la informació més rellevant per al model, reduint-ne la complexitat i millorant la capacitat de generalització. Per aquest motiu, hem decidit conservar només aquestes columnes, i els gràfics i comentaris finals que segueixen es basen en aquesta versió definitiva del model.

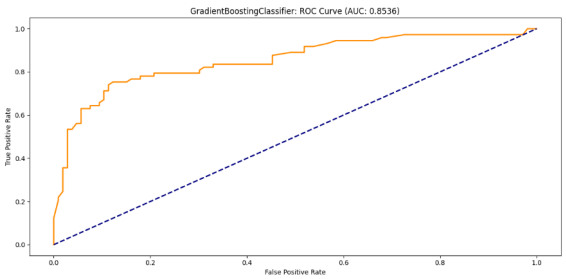

Figure 6. Corba AUC-ROC del Gradient Boosting

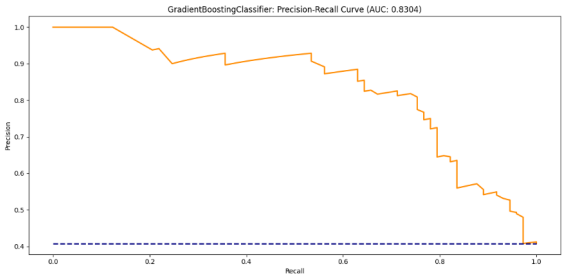

Figure 7. Corba Precision-Recall del Gradient Boosting

Accuracy	Precision	Recall	F1-score	AUC-ROC	AUC-PR
0.8044	0.8392	0.6438	0.7496	0.8535	0.8299

Table 3. Mètriques finals del model Gradient Boosting.

1. Avaluació de mètriques: La mètrica que menys ens agrada és el **recall**, ja que és força baixa (0.64) en comparació amb la **precision** (0.84). Això indica que el model tendeix a identificar bé els supervivents que prediu, però en deixa escapar alguns (falsos negatius). És a dir, és més "conservador" a l'hora de predir supervivència. Això pot ser conseqüència d'un desequilibri lleu entre classes i del fet que treballem amb poques mostres. Recordem que les mètriques s'han obtingut mitjançant *validació creuada*, i el conjunt de dades limitat incrementa la variabilitat dels resultats.

2. Interpretació global del rendiment: El **F1-score de 0.75** reflecteix un bon equilibri entre *precision* i *recall*, però no és un resultat excepcional. Esperàvem un valor una mica superior, però tenint en compte que el model està ben regularitzat i que s'ha evitat l'*overfitting*, considerem que el rendiment és satisfactori dins de les limitacions del dataset. El **ROC-AUC de 0.85** i el **PR-AUC de 0.83** mostren una bona capacitat de discriminació entre classes, amb corbes suaus i coherents (Figures 6 i 7).

3. Conclusions sobre el model: El **Gradient Boosting** s'ha mostrat com el model més robust i estable, amb una bona gestió del *bias-variance tradeoff*. Els hiperparàmetres triats (`learning_rate = 0.1`, `max_depth = 3`, `n_estimators = 201`) proporcionen un bon equilibri entre rendiment i generalització, evitant sobreajustar-se a les dades d'entrenament.

4. Marge de millora: Encara que el model és sòlid, hi ha espai per optimitzar-lo:

- **Feature engineering:** Analitzar i ajustar les variables d'entrada segons el model, com combinacions d'edat, títol i classe.
- **Hiperparàmetres:** Explorar altres paràmetres com `min_samples_leaf`, `subsample` o `max_features` per millorar generalització i *recall*.
- **Més dades o augment:** Ampliar el dataset o aplicar tècniques com *SMOTE* per equilibrar les classes i millorar el *recall*.
- **Models avançats:** Provar alternatives com **XGBoost** o **LightGBM** per possible millora de rendiment.

5. Conclusió final: Tot i les seves limitacions, el nostre model de **Gradient Boosting** aconsegueix un rendiment sòlid i estable, amb un bon balanç entre *precision* i *recall*. Les mètriques obtingudes mostren que el model generalitza correctament i no pateix sobreajustament greu, convertint-lo en una opció adequada per a problemes de classificació binària amb dades reduïdes, com el del Titanic.