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—An abstract is a brief summary that outlines the key aspects of a work. An example of a famous abstract is reproduced verbatim here for illustration purposes [**vaswani_attention_2017**]: The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 Englishto-German translation task, improving over the existing best results.

Keywords—*a, b, c, d*

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.

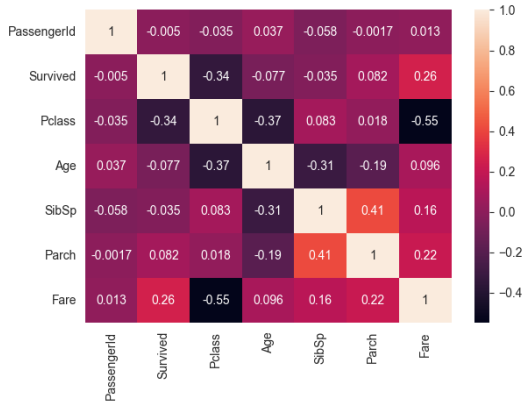

Figure 1. Matriu de correlacions

2. Preprocessing

Per a 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 forografía.

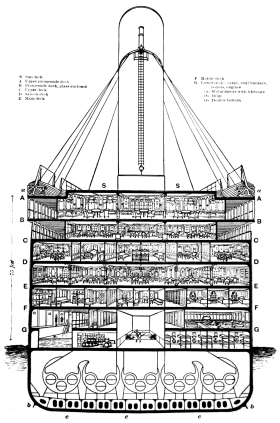

Figure 2. Decks

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 els 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 tíquets 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 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, creiem que amb aquestes és un bon moment per usar KNN per a trobar passatgers similars i poder assignar una edat als que no en tenim dades, 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 tíquet), 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.

Entrenem el model utilitzant regressió logística i 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.

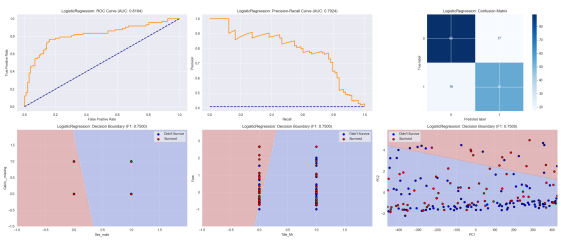

Figure 3. Gràfics

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 reobust, tot i tardar una mica més: Repeated Stratified K-Fold. L'implementem directament en la nostra class `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 hiperparàmetres o tenen risc d'overfit.

Hem provat a reduïr la dimensionalitat del model per a veure si obtenim bons resultats en dimensions menor que poguem visualitzar, aquests són alguns dels resultats que hem obtingut:

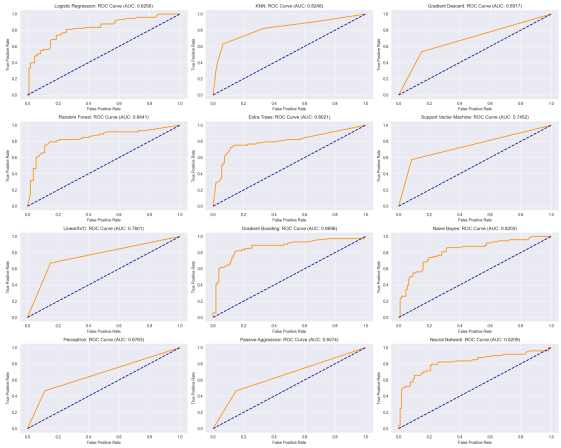

Figure 4. Corbes amb el dataset reduït

5. Anàlisi Final

RESULTATS DEL MODEL TRIAT AMB ELS MILLORS HIPERPARAMETRES

Preguntes: * Mostreu les corves ROC/PR (la que hagueu escollit en l'apartat 2) i interpreteu els resultats.

* Analitzeu en detall les diferents mètriques que trobeu adients i comenteu per sobre com podrieu fer servir aquest model en un futur. Això és el que es coneix com un cas d'ús.

* Com creieu que es podria millorar el vostre model?