Babacea Alexandru
311CC

Proiect PCLP3

Cerinta1:

Folosesc interquartile range pentru a identifica si elimina outlier-ele. Outlierii sunt cei cu valori mai mici decat Q1 - 1.5IQR sau mai mare decat Q3 + 1.5IQR, unde Q1 si Q3 sunt primul si al treilea percentile, iar IQR este diferenta dintre Q3 si Q1. Valorile care sunt în afara intervalului Q1 - 1.5IQR si Q3 + 1.5IQR sunt eliminate.

Cerinta2:

Calculez Z-score pentru fiecare observatie si elimin valorile care au un Z-score absolut mai mare decât un anumit prag. Z-score este numarul de deviatii standard fata de media setului de date. In cazul lui *Age*, valorile cu un Z-score mai mare de 1.9 sunt considerate outlier-e. Pentru *Fare* se elimina valorile cu Z-score mai mare ca 0.7 . Pentru aceste praguri rezultatele sunt asemanatoare cu cele ale functiei de la cerinta 1.

Cerintele 3 si 4:

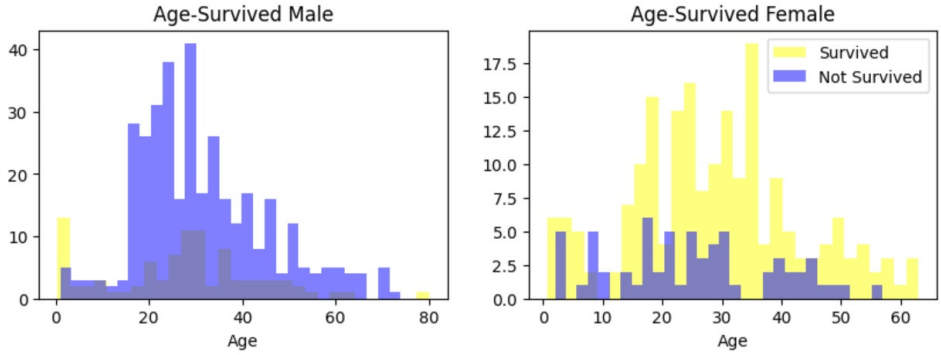
- **Analiza si prelucrarea setului de date de antrenament:**

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

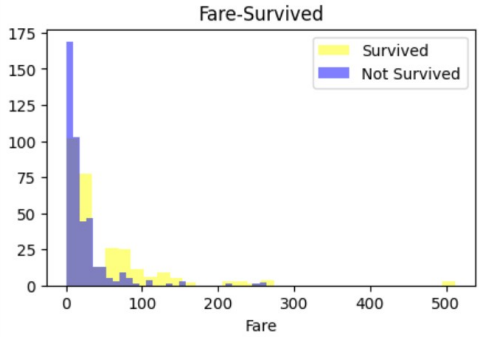
- Pentru toate datele in afara de *Age* (714 din 891), *Cabin* (204 din 891) si *Embarked* (889 din 891) setul de date are toate coloanele complete.
- Pentru *Age* completez valorile lipsa cu media valorilor existente. Media este 29.69911764705882.
- Din coloana *Cabin* se poate extrage puntea pe care se afla pasagerul, iar apoi elimin coloana Cabin, deoarece are prea multe valori lipsa si nu mai este utila. Coloana cu puntile se va numi *Deck*.
- Coloana de indici (*PassengerId*) nu aduce informatii utile, iar din coloana *Ticket* informatiile sunt greu de extras, asa ca le voi elimina pe amandoua.
- Coloana *SibSp* reprezinta numarul de frati/soti de la bordul Titanicului, iar *Parch* reprezinta numarul de parinti/copii de la bordul Titanicului, astfel are sens sa combinam cele 2 coloane intr-o coloana *Relatives*.

- La *Embarked* pot sa completez cu valoarea cea mai intalnita S (Southampton), deoarece sunt doar 2 valori lipsa.

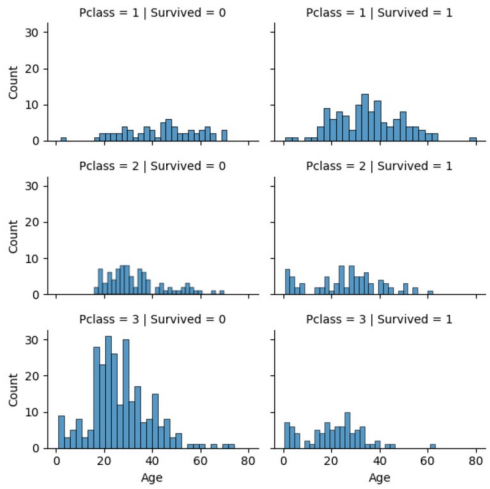
```
Embarked
S    644
C    168
Q     77
Name: count, dtype: int64
```



Exista o corelatie intre sexul persoanei si supravietuire, legata de varsta. Au sanse mai mari de supravietuire femeile intre 19 si 35 de ani. Au sanse mai mici de supravietuire barbatii intre 19 si 38 de ani. In consecinta voi grupa varstele pe clustere (voi discretiza datele).



In coloana *Fare* se pot observa outlieri evidenti care vor fi eliminati si, ca la varsta, voi discretiza rezultatul impartind datele pe clustere.



Din aceste grafice se poate observa importanta coloanei *Pclass* in predictii, in legatura cu varsta. Se observa ca cei la clasa a 3a au cele mai mici sanse de supravietuire, pe cand cei de la clasa 1 au sansele cele mai mari de supravietuire. Este utila o coloana care sa combine valorile din *Pclass* si din *Age*.

Modific coloana *Name* astfel incat sa pastrez in ea doar titlul persoanelor/formula de adresare. Fac asta cu ajutorul expresilor regulate. Rezultatul va arata astfel:

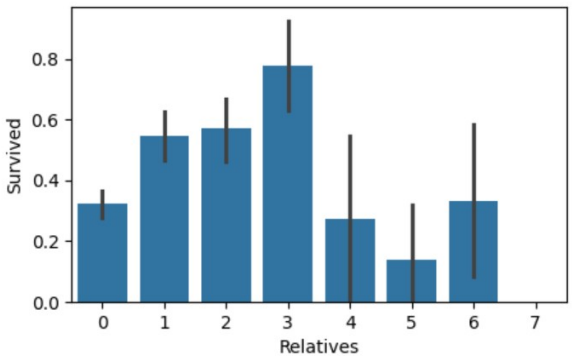
```
Name
Mr          517
Miss        182
Mrs         125
Master       40
Dr            7
Rev           6
Mlle          2
Major         2
Col           2
Countess      1
Capt          1
Ms            1
Sir           1
Lady          1
Mme           1
Don           1
Jonkheer      1
```

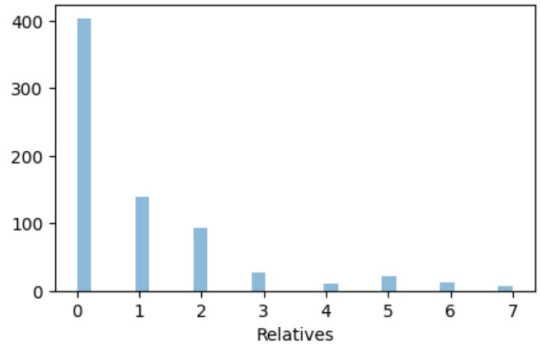
Pentru a mai restrange din valori, grupez Mlle(Mademoiselle) cu Ms(Miss) si cu (Ms) (toate inseamna domnisoara), grupez Mme(Madame) cu Mrs (ambele inseamna doamna), iar toate celelate valori care apar de mai putin de 10 ori sunt convertite in stringul ‘Rare’.

```
Name
Mr        517
Miss      185
Mrs       126
Master     40
Rare       23
Name: count, dtype: int64
```

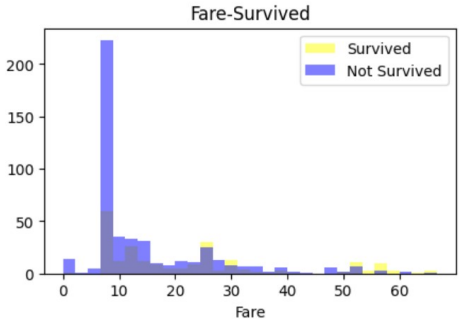
Din coloana *Cabin* pot extrage puntea persoanei si pot crea coloana *Deck*, deoarece pe coloana cabin este numarul puntii urmat de numarul cabinei. Deci, pot extarge puntea cu o extresie regulata. Inainte, totusi, trebuie sa completez in *Cabin* valorile lipsa. Pe pozitile goale pun N, urmat de 0 (N0) pentru a functiona expresia regulata si a ramane in coloana *Deck* doar valoarea N. Dupa aceasta pot sterge coloana *Cabin*. Coloana *Deck* va arata astfel:

```
Deck
N    687
C     59
B     47
D     33
E     32
A     15
F     13
G      4
T      1
```

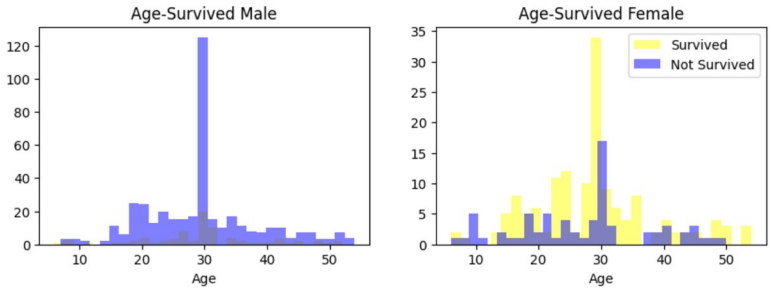




Asa cum am spus si mai sus, formez coloana *Relatives* prin adunarea coloanelor *SibSp* si *Parch*. Persoanele cu 3 rude au cele mai multe sanse de supravietuire, dar cele mai multe persoanele nu au rude.



Pentru a filtra coloana *Fare* (elimin outlierii) am folosit functia *z_score* si am eliminat valorile care sunt mai mici sau egale cu 0.7 . Am obtinut aceasta valoare constatand ca pentru ea se obtin rezultate foarte apropiat de cele ale functiei *iqr* si am pastrat-o deoarece ofera rezultate bune. Au ramas valori pentru pretul croazierei mai mici decat 70-75. Intradevar, si din graficul initial se observa ca valorile cele mai frecvente sunt concentrate sub 100.



Pentru a filtra coloana *Age* (elimin outlierii) am folosit functia *z_score* si am eliminat valorile care sunt mai mici sau egale cu 1.9 . Ca la *Fare*, am dedus aceasta valoare prin asemanarea cu *iqr* si am pastart-o, oferind rezultate bune. Din grafice se observa ca au ramas persoanele sub 55 de ani si, intradevar, se poate observa si din graficele initiale ca marea majoritate a persoanelor are sub 60 de ani.

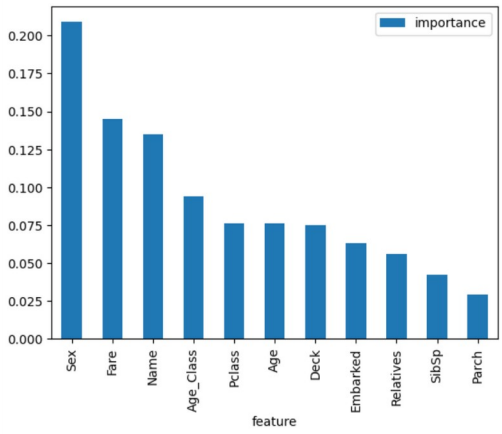
Convertesc coloanele *Sex*, *Embarked*, *Name* si *Deck* din valori categorice in valori numerice.

Impart varstele (*Age*) si preturile pentru croaziera (*Fare*) in 6 categorii. Daca as incerca o distributie pe intervale egale ar fi prea multe valori in unele intervale, iar in altele prea putine. Astfel, pentru fiecare din cele 2 coloane mentionate folosesc functia *qcut* din pandas pentru a impartii datele in mod cat de cat egal in intervale. Obs! Intervalele au mai fost ajustate fata de cele returnate de *qcut*, pentru a obtine o acuratete cat mai buna la final. Label-urile pentru fiecare interval de varsta / de pret sunt distribuite astfel:

```
Age
3    184
4    128
1    119
0    107
5     88
2     78
Name: count, dtype: int64
Fare
0    151
3    144
1    132
2    130
4     98
5     49
Name: count, dtype: int64
```

Creez coloana suplimentara *Age_Class* care combina valorile dintre clasa si varsta. Aceasta se dovedeste ca va avea o importanta semnificativa in antrenarea modelului.

Normalizez datele folosind *StandardScaler*.



- **<u>Analiza si prelucrarea setului de date de test:</u>**

Aleg una din cele 2 variante in functie de parametrul *scop* pe care il citesc.

1) Pentru cazul in care testez local, doresc impartirea setului din *train.csv* in 2 parti: 80% de antrenament si 20% de test. In acest caz setul de date de test nu mai trebuie prelucrat.

2) Pentru cazul in care dorez sa prezic pentru setul de date din *test.csv*, voi face asupra setului de date de test aceleasi operatii pe care le-am facut si asupra setului de antrenament, cu cateva mentiuni:

- Coloana *Embarked* este completa (nu mai trebuie completata), dar de data aceastea lipsesc valori din coloana *Fare*, din coloana *Age*, din coloana *Cabin.*
- Bineinteles, pentru test nu trebuie eliminati outlieri.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Pclass       418 non-null    int64
 2   Name         418 non-null    object
 3   Sex          418 non-null    object
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64
 6   Parch        418 non-null    int64
 7   Ticket       418 non-null    object
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object
 10  Embarked     418 non-null    object
dtypes: float64(2), int64(4), object(5)
```

- **<u>Antrenare cu RandomForestClassifier:</u>**

Setez parametrul random_state pentru a face tot timpul aceeasi predictie. Am ales sa il setez cu 3 dupa ce am testat toate valorile pana la 100.

Rezultate:

```
Acuratete: 0.8936170212765957

Raport:
              precision    recall  f1-score   support

           0       0.91      0.94      0.93       102
           1       0.83      0.77      0.80        39

    accuracy                           0.89       141
   macro avg       0.87      0.86      0.86       141
weighted avg       0.89      0.89      0.89       141
```

Rezultat de pe Kaggle:

Am preluat cateva idei legate de prelucrarea datelor de la urmatoarea adresa:
https://towardsdatascience.com/predicting-the-survival-of-titanic-passengers-30870ccc7e8