Cernea Mihnea-Ioan314CB

Proiect PCLP 3

Partea 1

Cerinta 1:

Acest script in Python utilizeaza biblioteca pandas pentru a analiza structura si calitatea datelor dintr-un fisier CSV numit train.csv. Scriptul realizeaza urmatoarele actiuni:

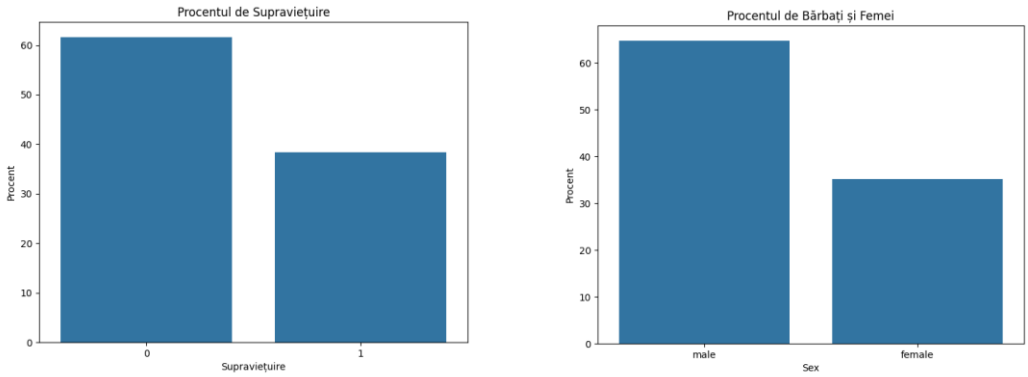
- Citirea datelor: Se incarca fisierul CSV intr-un DataFrame pentru a facilita analiza si manipularea datelor.
- Numarul de coloane si tipurile de date: Se afiseaza cate coloane sunt in DataFrame si ce tip de date are fiecare coloana (de exemplu, int64, float64, object etc.). Acest lucru este util pentru a intelege structura datelor si a identifica eventualele probleme (de exemplu, date numerice stocate ca siruri de caractere).
- Numarul de valori lipsa: Se afiseaza cate valori lipsa sunt in fiecare coloana. Aceasta este o parte cruciala a analizei de date, deoarece valorile lipsa pot afecta calitatea si rezultatele analizei.
- Numarul de linii: Se afiseaza cate linii (inregistrari) sunt in DataFrame. Acest lucru ofera o idee despre dimensiunea setului de date.
- Verificarea liniilor duplicate: Se verifica daca exista linii duplicate in DataFrame. Liniile duplicate pot distorsiona analizele statistice si modelele de invatare automata, deci este important sa fie identificate si tratate.

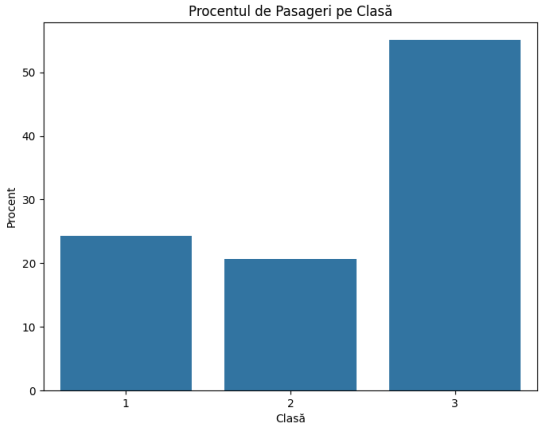
Cerinta 2:

Acest script in Python foloseste bibliotecile pandas, matplotlib, si seaborn pentru a analiza si vizualiza datele din setul train.csv. Mai jos este o explicatie detaliata a fiecarei parti din script:

- Citirea si analiza datelor:
    - Datele sunt citite din fisierul CSV si sunt stocate intr-un DataFrame pentru a facilita analiza.
    - Procentajele sunt calculate pentru a intelege distributia supravietuirii, clasei si sexului in setul de date.
- Vizualizarea datelor:
    - Graficele de tip barplot sunt utilizate pentru a vizualiza distributiile calculate. Aceste grafice ofera o reprezentare vizuala clara si usor de inteles a datelor.
    - Setarea dimensiunii figurii si adaugarea de titluri si etichete imbunatatesc claritatea si prezentarea graficelor.
    - Salvarea graficelor in fisiere PNG permite utilizarea acestora in rapoarte sau prezentari.

Cernea Mihnea-Ioan314CB



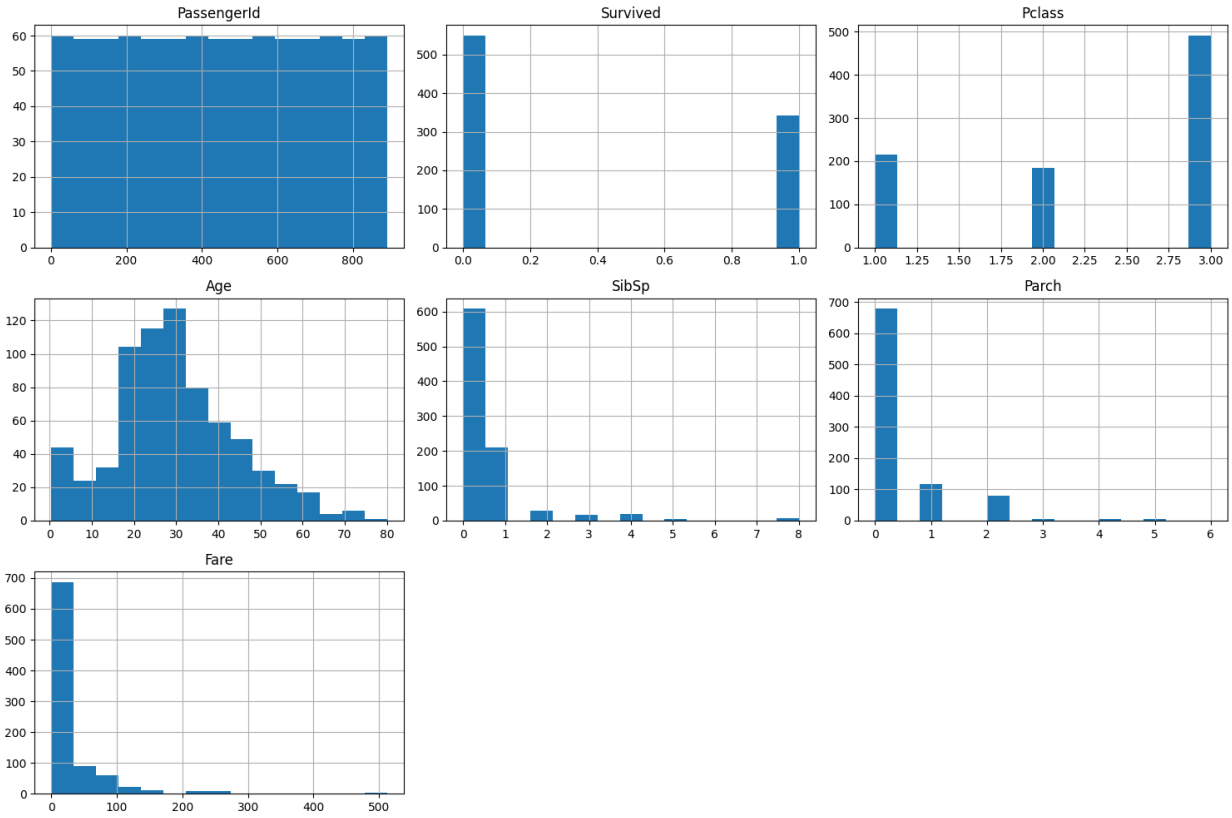


Cerinta 3:

Acest script in Python utilizeaza bibliotecile pandas si matplotlib pentru a genera histograme pentru toate coloanele numerice din setul de date train.csv. Mai jos este o explicatie detaliata a fiecarei parti din script:

- Citirea si pregatirea datelor:
  - Datele sunt citite din fisierul CSV si stocate intr-un DataFrame pentru a facilita analiza.
  - Se selecteaza doar coloanele numerice din DataFrame pentru a genera histograme, deoarece histogramele sunt relevante doar pentru date numerice.
- Generarea si salvarea histogramelor:
  - Histogramele sunt generate pentru a vizualiza distributia valorilor din fiecare coloana numerica. Acestea ofera o imagine clara a modului in care valorile sunt distribuite, evidentind caracteristici importante precum simetria, valori extreme (outliers), si gruparea datelor.
  - Setarea dimensiunii figurii si organizarea histogramelor intr-o grila ajuta la o prezentare clara si compacta a tuturor graficelor intr-o singura imagine.
  - Salvarea figurii intr-un fisier PNG permite utilizarea acesteia in rapoarte sau prezentari, facilitand partajarea rezultatelor analizei.

Cernea Mihnea-Ioan314CB



Cerinta 4:

Acest script în Python utilizează biblioteca pandas pentru a analiza datele din fișierul train.csv, identificând coloanele cu valori lipsă și calculând procentul acestor valori pentru fiecare clasă de supraviețuire (supraviețuit sau nu). Mai jos este o explicație detaliată a fiecărei părți din script:

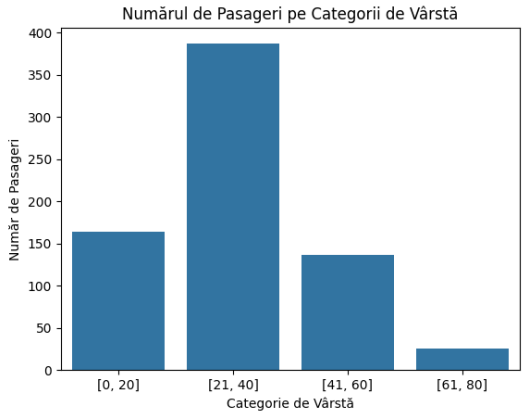
- Identificarea valorilor lipsă:
  - Se calculează numărul de valori lipsă pentru fiecare coloană și se selectează coloanele care au cel puțin o valoare lipsă. Aceasta permite identificarea rapidă a problemelor de completitudine a datelor.
- Analiza procentuală a valorilor lipsă:
  - Se calculează și se afișează procentul de valori lipsă pentru fiecare clasă de supraviețuire (supraviețuit sau nu) pentru fiecare coloană cu valori lipsă. Această analiză poate dezvălui diferențe semnificative între clase, care ar putea indica un bias sau o problemă specifică cu colectarea datelor pentru anumite grupuri de pasageri.

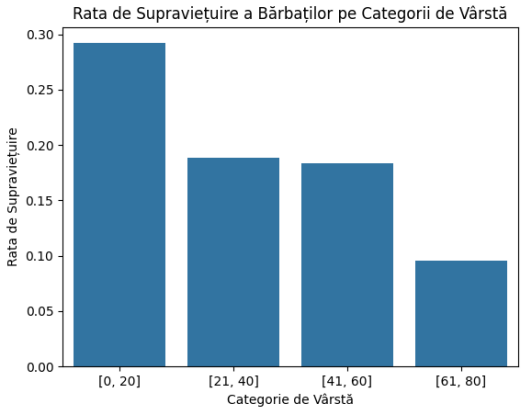
Cerinta 5si 6:

Acest script in Python are ca scop analizarea datelor din setul de date Titanic referitoare la varsta si rata de supravietuire a barbatilor pe categorii de varsta. Iata o explicatie detaliata a fiecarei parti a scriptului:

- Definirea categoriilor de varsta:
  - Se calculeaza valoarea maxima a varstei din coloana 'Age' folosind metoda max().

  - Se utilizeaza functia pd.cut() pentru a imparti valorile varstelor in intervale prestabilite si pentru a le eticheta cu numele corespunzator al intervalului. Intervalul maxim este inclus de la 61 la valoarea maxima a varstei.
  - Rezultatul este adaugat ca o noua coloana in DataFrame sub numele 'Age_Category'.
  - Calculul numarului de pasageri pentru fiecare categorie de varsta folosind metoda value_counts() pentru coloana 'Age_Category'.
  - Afisarea si salvarea unui grafic de tip barplot pentru a vizualiza numarul de pasageri pe categorii de varsta folosind sns.countplot().
- Calculul ratei de supravietuire a barbatilor pentru fiecare categorie de varsta:
  - Selectarea datelor pentru barbati folosind un filtru pe coloana 'Sex'.
  - Se grupeaza datele in functie de categoriile de varsta si se calculeaza media supravietuirii pentru fiecare categorie folosind groupby() si mean().
- Afisarea si salvarea unui grafic de tip barplot pentru a vizualiza rata de supravietuire a barbatilor pe categorii de varsta folosind sns.barplot().



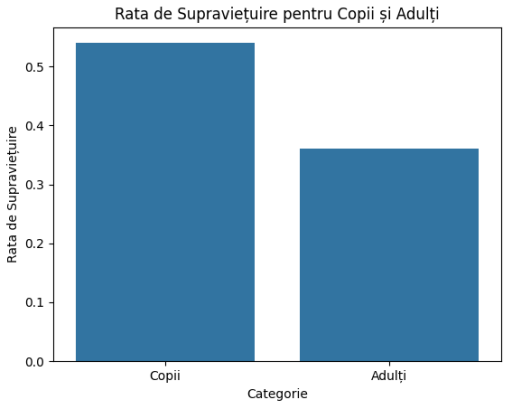


Cerinta 7:

Acest script Python analizeaza rata de supravietuire pentru copii si adulti din setul de date Titanic. Iata o explicatie detaliata a fiecarei parti a scriptului:

- Definirea copiilor ca fiind persoanele cu varsta sub 18 ani prin adaugarea unei noi coloane numite 'IsChild' in DataFrame. Aceasta coloana va avea valoarea True pentru pasagerii cu varsta sub 18 ani si False pentru cei cu varsta de 18 ani sau peste, utilizand o expresie booleana df['Age'] < 18.
- Calculul procentului de copii la bord, folosind metoda mean() pentru a calcula media valorilor din coloana 'IsChild' si inmultind rezultatul cu 100 pentru a obtine procentul.
- Calculul ratei de supravietuire pentru copii si adulti:
  - Se selecteaza datele pentru copii si adulti folosind filtrul df[df['IsChild'] == True] si df[df['IsChild'] == False], respectiv.
  - Pentru fiecare grup, se calculeaza media coloanei 'Survived' folosind metoda mean().
- Crearea unui grafic de tip barplot pentru a vizualiza rata de supravietuire a copiilor si adultilor:
  - Se defineste lista survival_rates care contine ratele de supravietuire calculate anterior.
  - Se defineste lista labels care contine etichetele 'Copii' si 'Adulti'.

  - Se utilizeaza sns.barplot() pentru a crea graficul, specificand valorile de pe axa x si y, respectiv etichetele pentru axe.
- Salvarea graficului in fisierul 'Analiza_rata_de_supravietuire_copii_adulti.png' folosind plt.savefig().



Cerinta 8:

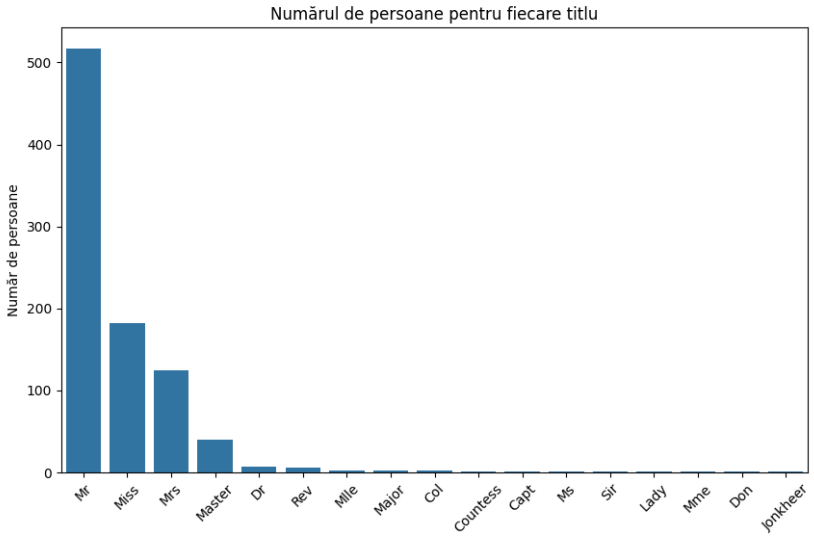
Acest script Python folosește două funcții pentru a completa valorile lipsă din setul de date Titanic și apoi salvează DataFrame-ul completat într-un nou fișier CSV. Iată o explicație detaliată a fiecărei părți a scriptului:

- Definirea a două funcții pentru completarea valorilor lipsă:
  - Funcția fill_missing_numerical: Completează valorile lipsă pentru o coloană numerică (cum ar fi 'Age') cu media valorilor din acea coloană, grupate după o altă coloană dată (cum ar fi 'Survived').
  - Funcția fill_missing_categorical: Completează valorile lipsă pentru o coloană categorială (cum ar fi 'Embarked') cu cea mai frecventă valoare din acea coloană, grupată după o altă coloană dată (cum ar fi 'Pclass').
- Apelul funcțiilor definite anterior pentru completarea valorilor lipsă în DataFrame-ul df:
  - Pentru coloana 'Age', valorile lipsă sunt completate pe baza mediei vârstelor pasagerilor din aceeași categorie de supraviețuire (Survived).
  - Pentru coloana 'Embarked', valorile lipsă sunt completate pe baza celei mai frecvente valori de îmbarcare (Embarked) din aceeași clasă de pasageri (Pclass).
  - Pentru coloana 'Cabin', valorile lipsă sunt completate pe baza celei mai frecvente valori de cabină (Cabin) din aceeași clasă de pasageri (Pclass).
- Salvarea DataFrame-ului completat în fișierul 'train_filled.csv' folosind metoda to_csv() cu parametrul index=False, pentru a evita salvarea indexului DataFrame-ului în fișierul CSV.

Cerinta 9:

Acest script Python utilizează bibliotecile pandas, seaborn și matplotlib.pyplot pentru a analiza titlurile de nobilime și pentru a verifica dacă acestea corespund sexului persoanelor respective în setul de date Titanic. Iată o explicație detaliată a fiecărei părți a scriptului:

- Extragem titlurile de nobilime din coloana 'Name' folosind metoda str.extract(), care utilizează o expresie regulată pentru a identifica secvențele de caractere care corespund unui anumit pattern. Rezultatul este stocat într-o nouă coloană numită 'Title'.
- Creăm un dicționar title_gender_mapping care mapează fiecare titlu la sexul corespunzător. Această asociere se bazează pe cunoștințele convenționale despre titlurile de nobilime și sexul asociat acestora.
- Adăugăm o nouă coloană în DataFrame, numită 'Gender', care mapează titlurile din coloana 'Title' la sexul corespunzător folosind dicționarul title_gender_mapping.
- Calculăm numărul de persoane pentru fiecare titlu folosind metoda value_counts() și stocăm rezultatul în variabila title_counts.
- Reprezentăm grafic numărul de persoane pentru fiecare titlu folosind sns.countplot(), care afișează un grafic de bare pentru distribuția fiecărui titlu. Folosim ordinea descrescătoare a titlurilor în funcție de numărul de apariții pentru a facilita înțelegerea rezultatelor. Etichetele pe axa x sunt rotite cu 45 de grade pentru a asigura o mai bună vizibilitate a titlurilor.
- Salvăm graficul sub forma unui fișier de imagine PNG folosind plt.savefig().



Cerinta 10:

Acest script utilizează bibliotecile pandas, matplotlib.pyplot și seaborn pentru a analiza influența stării de a fi singur asupra șanselor de supraviețuire și relația dintre tarif, clasă și supraviețuire pentru primele 100 de înregistrări din setul de date Titanic. Iată o explicație a fiecărei părți a scriptului:

- Crearea coloanei 'IsAlone':
  - Se adaugă o coloană nouă numită 'IsAlone' care indică dacă un pasager este singur sau nu. Această coloană este calculată pe baza informațiilor din coloanele 'SibSp' (numărul de frați și soții la bord) și 'Parch' (numărul de părinți și copii la bord).
- Histograma pentru influența singurătății asupra supraviețuirii:
  - Se trasează o histogramă utilizând sns.histplot() pentru a vizualiza influența faptului de a fi singur asupra șanselor de supraviețuire. Histograma este colorată în funcție de

supraviețuire, cu două culori distincte pentru pasagerii care sunt singuri și cei care nu sunt. Etichetele de pe axa x sunt setate pentru a afișa "Nu" și "Da" în loc de 0 și 1.

- Vizualizarea relației dintre tarif, clasă și supraviețuire:
    - Se selectează primele 100 de înregistrări din DataFrame și se trasează un stripplot folosind sns.stripplot(). Acest grafic arată relația dintre tarif, clasă și supraviețuire pentru primele 100 de înregistrări. Jittering-ul este activat pentru a evita suprapunerile, iar punctele sunt separate în funcție de starea de supraviețuire utilizând argumentul dodge=True.
    - Salvarea imaginilor:
- Ambele grafice sunt salvate ca fișiere de imagine PNG folosind plt.savefig().