**MINISTERUL EDUCAŢIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Programul de studii: Tehnologia informației**



**RAPORT**

**Disciplina „Inteligenta Artificiala”**

**Tema: Regresie liniară**

|  |  |  |
| --- | --- | --- |
| **Student(ă):** | **\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Vlași**ț**chi Ștefan , TI-216** |
|  |  |  |
| **Coordonator universitate:** | **\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Rusu , asist.univ.** |

**Chișinău, 2024**

**Sarcini practice:**

1. **Importarea** **datelor**

**Pasul 1:** Importați panda, numpy, matplotlib și seaborn. Apoi setați %matplotlib inline (veți importa apoi sklearn după necesitate).

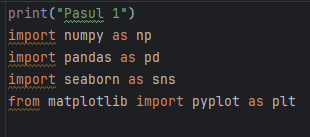


Figura 1 – Importarea librariilor

1. **Obținerea datelor**

Vom lucra cu fișierul Ecommerce Customers CSV de la companie. Are informații despre clienți, cum ar fi e-mail, adresă și avatar de culoare. Apoi are și coloane cu valori numerice: - Avg. Session Length: durata medie a sesiunilor de consiliere stil în magazin. - Time on App: timpul mediu petrecut pe aplicație în minute. - Time on Website: timpul mediu petrecut pe site în minute - Length of Membership: de câți ani este membru clientul.

**Pasul 2:** Citiți fișierul Ecommerce Customers CSV ca DataFrame numit customers.



Figura 2.1 – Citirea fișierului CSV într-un DataFrame

**Pasul 3:** Citiți primele linii (head) despre clienți și aflați informații despre ei folosinf funcțiile info() și describe().

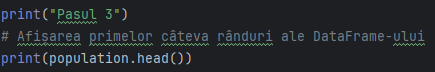


Figura 2.2 – Importarea librariilor

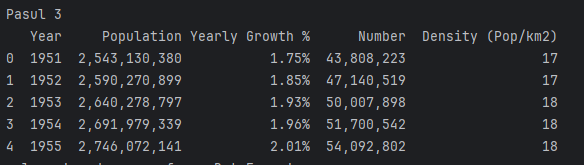


Figura 2.3 – Date experimentale

1. **Analiza exploratorie a datelor**

Pentru restul exercițiului vom folosi doar datele numerice ale fișierului csv. Utilizați seaborn pentru a crea o diagramă comună (jointplot) pentru a compara coloanele Time on Website și Yearly Amount Spent. Are sens corelația?

1. **Pasul 4:** Faceți același lucru, dar cu coloana Year si Density

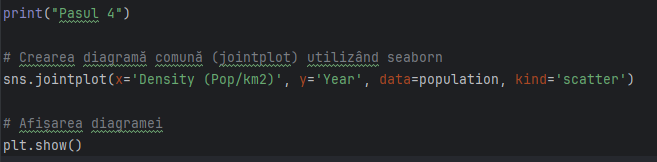


Figura 4.1 – Creare diagrama cu seaborn

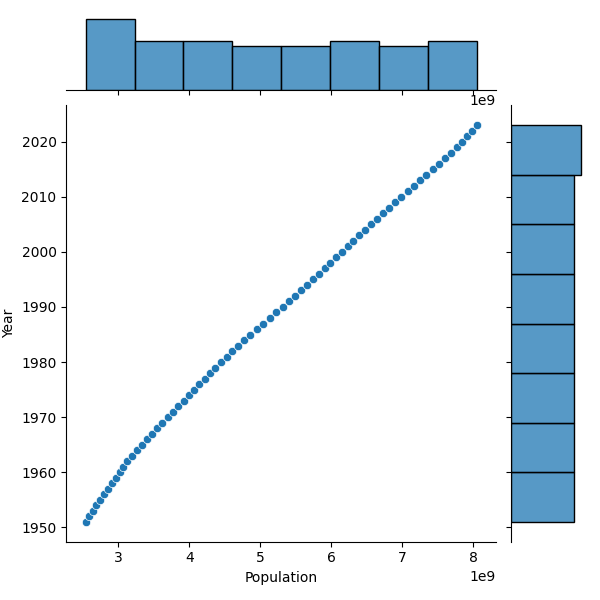


Figura 4.2 –Diagrama Year/Population

**Pasul 5:** Folosiți jointplot() pentru a crea o diagramă 2D ce compară Time on App și Length of

Membership.

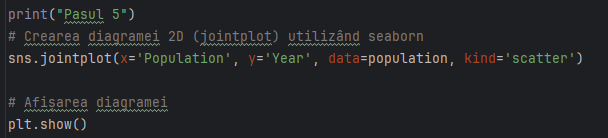


Figura 4.3 – Creare diagrama 2D cu seaborn

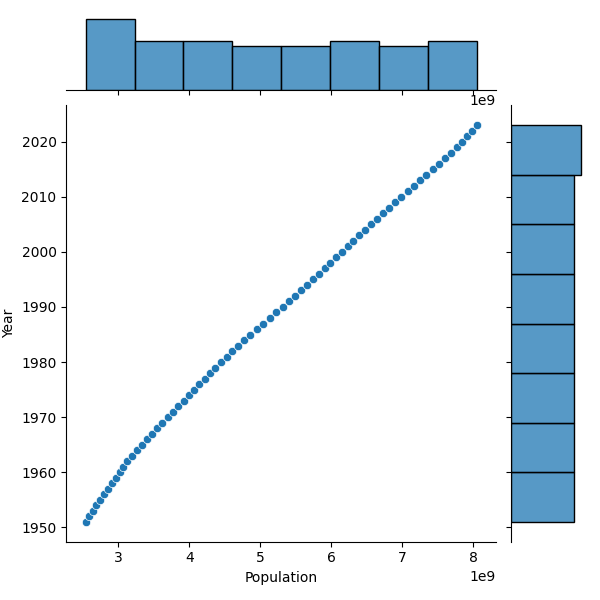


Figura 4.4 – Figura 4.2 –Diagrama Year/Population 2D

**Pasul 6:** Să explorăm aceste tipuri de relații în întregul set de date. Utilizați pairplot pentru a recrea graficul de mai jos. (Nu vă faceți griji pentru culori)

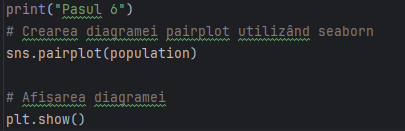


Figura 4.5 –Crearaea diagramei pairplot utilizand seaborn

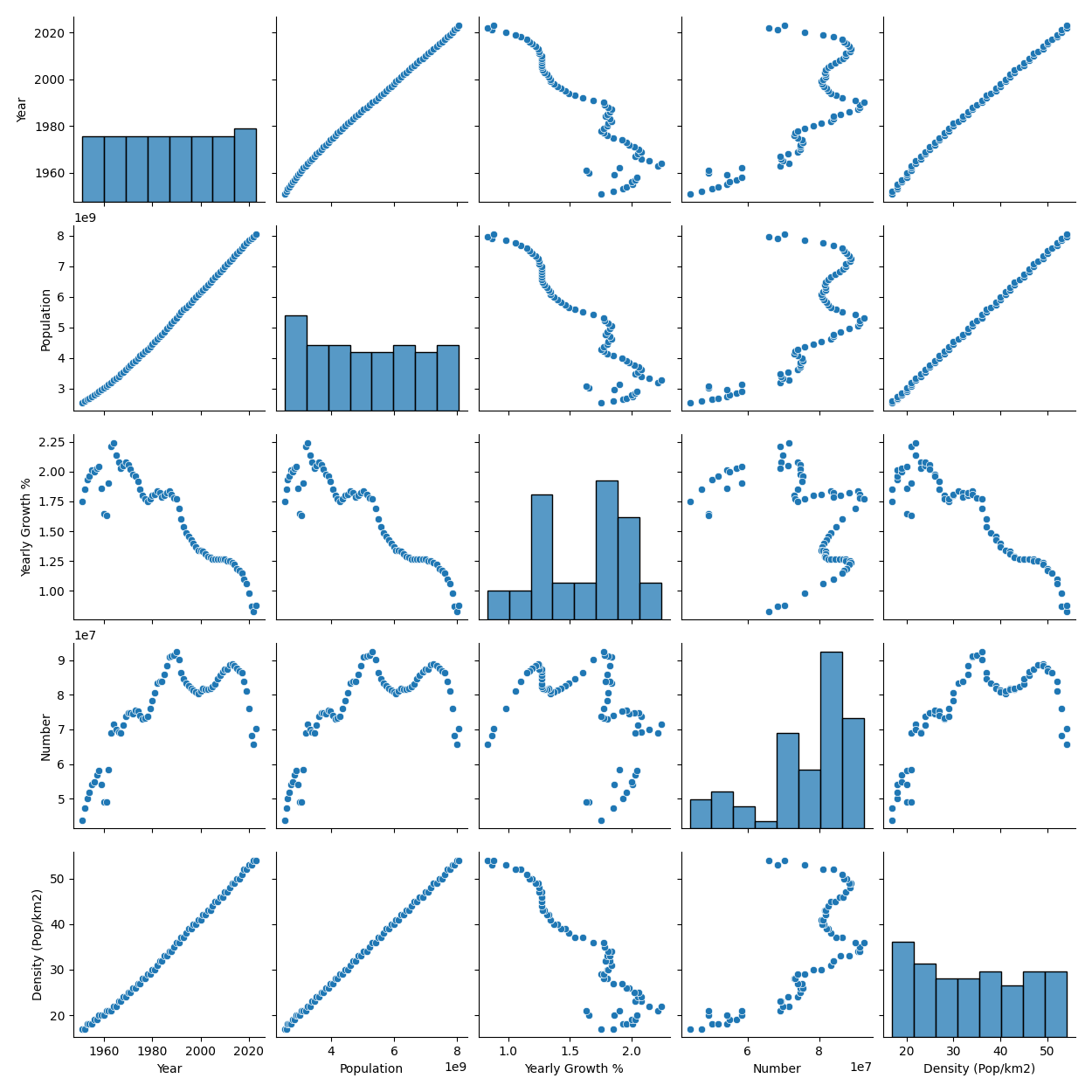


Figura 4.6 –Crearaea diagramei pairplot

**Pasul 7:** Creați un model liniar grafic (folosind Implot din seaborn) al datelor: Population vs Yearly Growth

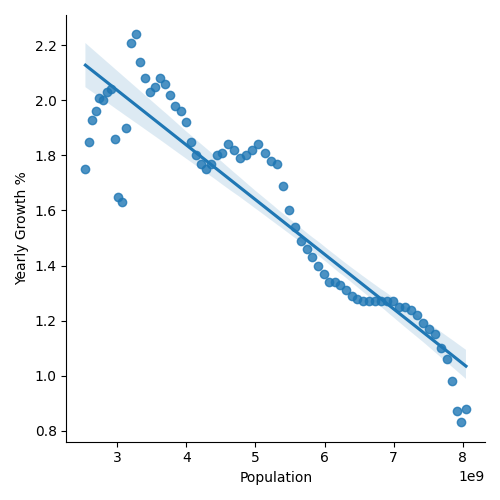


Figura 4.7 –Diagrama lineara grafica

1. **Date de instruire și testare**

Să împărțim datele în seturi de antrenament și de testare.

**Pasul 8:** Setați o variabilă X egală cu caracteristicile numerice ale clienților și o variabilă y egală cu coloana Yearly Amount Spent.

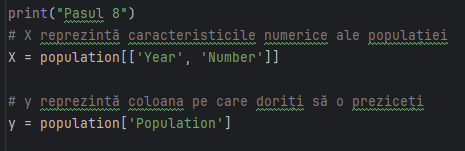


Figura 5.1 – Caracteristice numerice

**Pasul 9:** Utilizați model\_selection.train\_test\_split din sklearn pentru a împărți datele în seturi de antrenament și de testare. Setați test\_size=0,3 și random\_state=101.

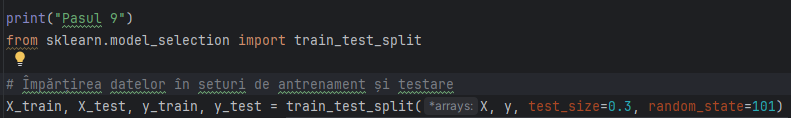


Figura 5.5 – Setarea datelor

1. **Antrenarea modelului**

Acum este timpul să ne instruim modelul pe datele noastre de antrenament!

**Pasul** **10:** Importați LinearRegression din sklearn.linear\_model.



Figura 6.1 – Importarea librariei

**Pasul** **11:** Creați o instanță a unui model LinearRegression() numit lm.

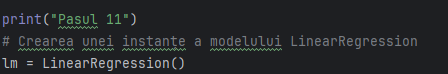


Figura 6.2 – Crearea unei instante

**Pasul** **12:** Antrenați lm pe datele de antrenament. LinearRegression(copy\_X=True, fit\_i ntercept=True, n\_jobs=1, normalize=False)

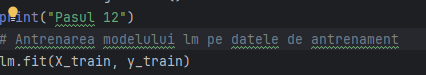


Figura 6.3 – Antrenarea modelului de date

Pasul 13: Imprimați coeficienții modelului.

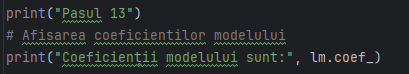


Figura 6.4 – Afisarea coeficientilor

1. **Prezicerea datelor de testare**

Să evaluăm performanța modelului prin prezicerea valorilor testului!

**Pasul 14:** Utilizați lm.predict() pentru a prezice setul X\_test de date.

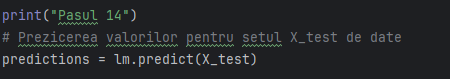


Figura 7.1 – Precizara valorilor

**Pasul 15:** Creați o diagramă de dispersie a valorilor reale de test față de valorile prezise.

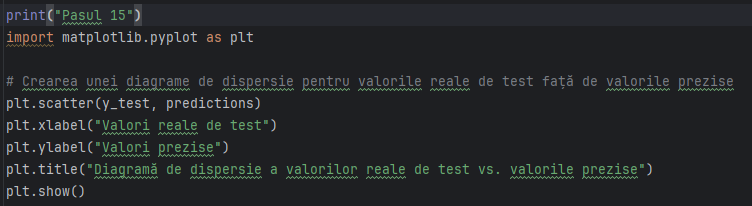


Figura 7.3 – Crearea diagramei

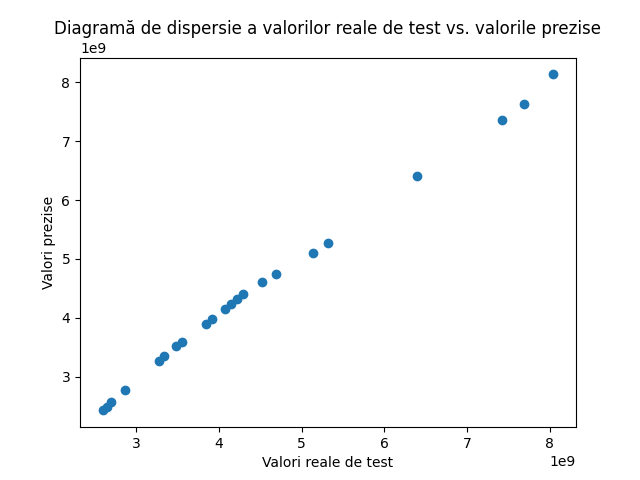


Figura 7.4 – Diagrama dispersiei

1. **Evaluarea** modelului

Să evaluăm performanța modelului nostru calculând suma reziduală a pătratelor și scorul de vari anță (R^2).

**Pasul 16:** Calculați eroarea medie absolută, eroarea medie pătratică și eroarea medie pătratică. Consultați prelegerea sau Wikipedia pentru formule.

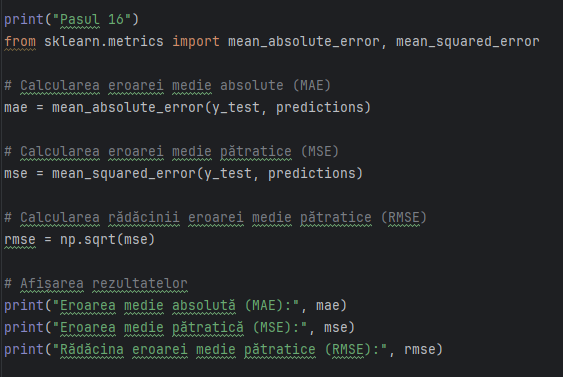


Figura 8.1 – Calcularea erorilor

1. Reziduuri

Să explorăm rapid reziduurile pentru a ne asigura că totul a fost în regulă cu datele noastre. **Pasul 17:** Trasați o histogramă a reziduurilor și asigurați-te că are o distribuție normală. Utilizați fie seaborn distplot, fie doar plt.hist().

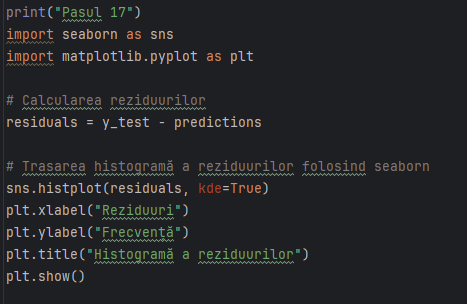


Figura 9.1 – Crearea histogramei

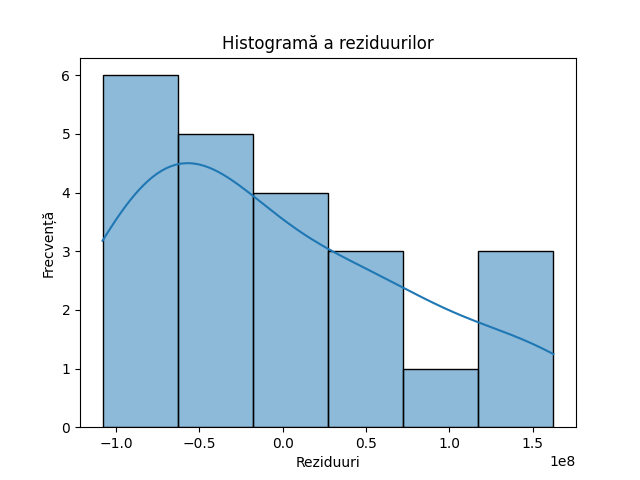


Figura 9.2 – Histogerama a rezidurilor

**Pasul 18:** Recreați setul de date de mai jos.

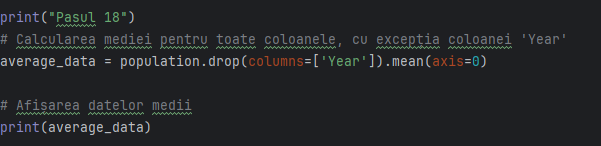


Figura 9.3 – Calcularea mediei

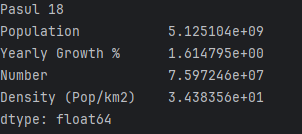


Figura 9.4 – Mediile calculate

**Pasul 19:** Răspundeți la întrebare: Cum puteți interpreta acești coeficienți?

Interpretarea coeficienților în contextul unei regresii liniare este esențială pentru înțelegerea relațiilor dintre variabilele implicate. În regresia liniară, fiecare coeficient este asociat cu o caracteristică specifică sau o variabilă independentă.

Dacă coeficientul este pozitiv, acest lucru indică o corelație pozitivă între variabila independentă și variabila dependentă. Cu alte cuvinte, o creștere în variabila independentă este asociată cu o creștere în variabila dependentă și invers.

Dacă coeficientul este negativ, acest lucru indică o corelație negativă între variabila independentă și variabila dependentă. Asta înseamnă că o creștere în variabila independentă este asociată cu o scădere în variabila dependentă și invers.

Interpretarea magnitudinii coeficienților este, de asemenea, importantă. Cu cât magnitudinea unui coeficient este mai mare, cu atât este mai semnificativă contribuția variabilei respective la predicția variabilei dependente.

**Pasul 20:** Răspundeți la întrebare: Credeți că compania ar trebui să se concentreze mai mult pe aplicația mobilă sau pe site-ul lor web?

Răspunsul la această întrebare depinde de mai mulți factori, inclusiv de obiectivele strategice ale companiei, preferințele clienților, analiza performanței actuale a aplicației mobile și a site-ului web, și alte aspecte. În general, este important să se evalueze și să se compare beneficiile și costurile asociate dezvoltării și îmbunătățirii aplicației mobile și site-ului web.

Unele aspecte pe care le puteți lua în considerare includ:

Utilizarea actuală și potențială a aplicației mobile și a site-ului web de către clienți.

Feedback-ul și preferințele clienților în ceea ce privește experiența utilizatorului și funcționalitățile dorite.

Analiza datelor despre utilizare și performanță, cum ar fi ratele de conversie, timpul petrecut pe platforme, feedback-ul utilizatorilor etc.

Investițiile necesare pentru dezvoltarea și întreținerea aplicației mobile versus a site-ului web.

**Concluzie:**

Până în prezent, am utilizat diverse metode pentru manipularea și analiza datelor din setul nostru de date. Iată o concluzie referitoare la metodele de operare cu aceste date:

În prima etapă, am importat bibliotecile necesare pentru lucrul cu datele noastre, inclusiv `pandas`, `numpy`, `matplotlib` și `seaborn`, pentru a efectua analize și vizualizări.

Am citit datele noastre dintr-un fișier CSV utilizând funcția `pd.read\_csv()` din biblioteca pandas. Aceasta ne-a permis să încărcăm datele într-un DataFrame pandas pentru a le putea analiza și manipula ulterior.

Am folosit diverse metode pentru a explora și înțelege setul nostru de date. Aceste metode includ afișarea primelor rânduri ale datelor cu `head()`, obținerea informațiilor despre setul de date cu `info()`, precum și calculul unor statistici de bază cu `describe()`.

Am folosit biblioteca `seaborn` pentru a crea diverse vizualizări, cum ar fi diagramele de dispersie (scatter plots) și diagramele comune (joint plots), pentru a explora relațiile dintre diferitele caracteristici ale datelor noastre.

Am efectuat operațiuni de curățare și prelucrare a datelor, inclusiv eliminarea valorilor lipsă, conversia datelor în tipuri adecvate (de exemplu, convertirea datelor de procent în float), și ajustarea intervalului de valori pentru o vizualizare mai precisă a datelor.

În ansamblu, aceste metode ne-au permis să explorăm, să analizăm și să vizualizăm datele noastre într-un mod eficient și informativ. Prin utilizarea acestor tehnici, putem obține o mai bună înțelegere a datelor și a relațiilor dintre diferitele lor caracteristici, ceea ce ne poate ajuta să luăm decizii mai informate în viitorul analizei noastre.