

**Analiza si Exploatarea Datelor**

**Examen**

**Profesor Maican Catalin**

**Membri:**

**Ivan Cosmin Denis**

**Iov Tudor**

**Olariu Oana**

**Manta Alexia**

**BRAȘOV, 2024**

*Universitatea Transilvania din Brașov*

*Facultatea de Științe Economice și Administrarea Afacerilor*

*Program de studii:Informatica Economica*

**Regression models to predict price/sales**

**Profesor Maican Catalin**

**Membri:**

**Ivan Cosmin Denis**

**Iov Tudor**

**Olariu Oana**

**Manta Alexia**

**BRAȘOV, 2024**

Aflam structura setului de date si informatii despre acesta

print(df.info())

Rezultatul acestei functii este:

RangeIndex: 1303 entries, 0 to 1302

Data columns (total 13 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 laptop\_ID 1303 non-null int64

1 Company 1303 non-null object

2 Product 1303 non-null object

3 TypeName 1303 non-null object

4 Inches 1303 non-null float64

5 ScreenResolution 1303 non-null object

6 Cpu 1303 non-null object

7 Ram 1303 non-null object

8 Memory 1303 non-null object

9 Gpu 1303 non-null object

10 OpSys 1303 non-null object

11 Weight 1303 non-null object

12 Price\_euros 1303 non-null float64

dtypes: float64(2), int64(1), object(10)

memory usage: 132.5+ KB

Aceasta functie ne indica cate coloane avem, numele acestora, numarul de inregistrari nenule ale fiecarei coloane si tipul de date.

print(df.describe())

laptop\_ID Inches Price\_euros

count 1303.000000 1303.000000 1303.000000

mean 660.155794 15.017191 1123.686992

std 381.172104 1.426304 699.009043

min 1.000000 10.100000 174.000000

25% 331.500000 14.000000 599.000000

50% 659.000000 15.600000 977.000000

75% 990.500000 15.600000 1487.880000

max 1320.000000 18.400000 6099.000000  
  
Aceasta functie ne arata numarul, media, deviatiile standard, minimul, maximul, cvartile pentru toate valorile numerice din setul de date.

Aceste 2 functii sunt folosite pentru a determina structura setului de date.

Stergem valorile nule

df.dropna(inplace=True)

Inlocuim NaN values cu valoarea cea mai frecventa care apare in coloana PanelType(mode)

df['PanelType'] = df['PanelType'].fillna(df['PanelType'].mode()[0])

Cand am facut coloana “PanelType” am transformat “ScreenResolution” in “PanelType” si “Resolution” si am inlocuit valorile nule cu cea mai frecventa valoare care apare in aceasta coloana si anume “Full HD”.

Înlocuim valorile NaN cu 0

df.fillna(0, inplace=True)

Pentru laptopurile cu 2 discuri de memorie am mai creat o coloana “MemoryBonus” in care puneam memoria in plus. Aici au aparut mai multe valori nule (NaN) pe care le-am inlocuit cu 0 pentru a le aduna in coloana “MemoryTotal”.

Verificam daca exista valori duplicate si in cazul in care exista le stergem(in cazul nostu avem 0 valori duplicate)

valori\_duplicate = df.duplicated()

print("Numar de valori duplicate:", valori\_duplicate.sum())

Rezultatul este:

Numar de valori duplicate: 0

Numarul total de inregistrari

numar\_inregistrari = df["laptop\_ID"]

print ("Numar inregistrari: ", numar\_inregistrari.count())

Rezultatul este:

Numar inregistrari: 1303

In continuare am curatat setul de date pentru a putea face analiza exploratorie si regresia

Calculează media, deviația standard, mediana, min, max si cvartilele pentru coloana "laptop\_ID"

df\_laptop\_ID = pd.DataFrame({

'Mean\_laptop\_ID': [df['laptop\_ID'].mean()],

'Std\_laptop\_ID': [df['laptop\_ID'].std()],

'Min\_laptop\_ID': [df['laptop\_ID'].min()],

'Quantile\_25%\_laptop\_ID': [df['laptop\_ID'].quantile(0.25)],

'Quantile\_50%\_laptop\_ID': [df['laptop\_ID'].quantile(0.5)],

'Quantile\_75%\_laptop\_ID': [df['laptop\_ID'].quantile(0.75)],

'Max\_laptop\_ID': [df['laptop\_ID'].max()]

})

Afișăm rezultatul

print("Laptop\_ID: ", df\_laptop\_ID)

In contextul calculului statisticii pentru coloana "laptop\_ID", termenul "media" se refera la media aritmetica a valorilor din acea coloana. Media aritmetica se calculeaza adunand toate valorile din coloana si impartind suma rezultata la numarul total de valori.

Deviatia standard este o măsura de dispersie sau variabilitate într-un set de date. Deviatia standard reprezinta cat de mult se abat valorile individuale din acea coloana fata de media aritmetica a coloanei.

Min inseamna valoarea minima a coloanei respective din setul de date in cazul codului de mai sus minimul din coloanal “laptop\_ID”.

Max inseamna valoarea maxima a coloanei respective din setul de date in cazul codului de mai sus maximul din coloanal “laptop\_ID”.

Cvartila este un concept statistic care descrie pozitia relativa a unui set de date in cadrul distributiei sale. Se refera la valorile care impart un set de date in patru parti egale, fiecare reprezentand un sfert al setului de date. Aceste valori sunt cunoscute sub numele de cvartile si sunt utilizate pentru a intelege distributia si variabilitatea datelor.

Cvartila de 50% inseamna mediana.

Am repetat acelasi proces pentru toate coloanele numerice din df: “Inches”,

”Price\_euros” “GHz”, “RamGB”, “WeightKG”, “MemoryGB”, ”MemoryBonus”,

“MemoryTotal”

Analiza exploratorie

Cel mai folosit Panou este: Full HD

Cel mai folosit Cpu este: Intel Core i5 7200U

Cel mai folosit producator de Gpu este: Intel

Cel mai folosit Gpu este: HD Graphics 620

Cea mai folosita rezolutie este: 1920x1080

Cel mai folosit produs este: XPS 13

Cea mai populara companie este: Dell

Cel mai folosit sistem de operare este: Windows 10

Cea mai populara diagonala este: 15.6

Cel mai folosit Panou este: Full HD

Cel mai folosit Cpu este: Intel Core i5 7200U

Cel mai folosit producator de Gpu este: Intel

Cel mai folosit Gpu este: HD Graphics 620

Cea mai folosita rezolutie este: 1920x1080

Cel mai folosit produs este: XPS 13

Cea mai populara companie este: Dell

Cel mai folosit sistem de operare este: Windows 10

Cea mai populara diagonala este: 15.6

Cel mai nefolosit Panou este: Touchscreen / Quad HD+

Cel mai nefolosit Cpu este: AMD FX 8800P

Cel mai nefolosit producator de Gpu este: ARM

Cel mai nefolosit Gpu este: GeForce 920

Cea mai nefolosita rezolutie este: 2736x1824

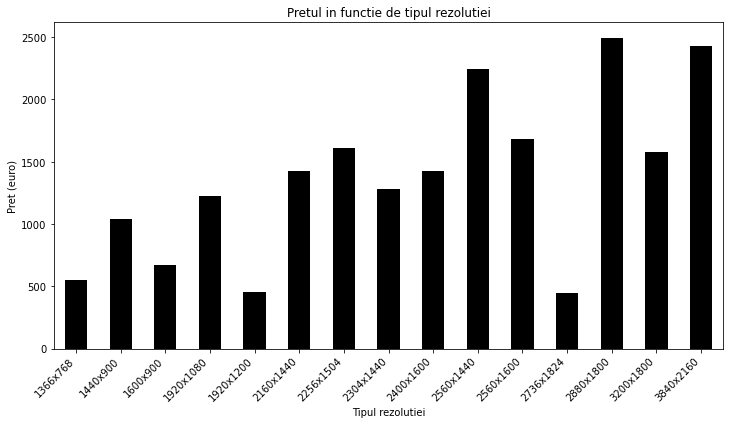
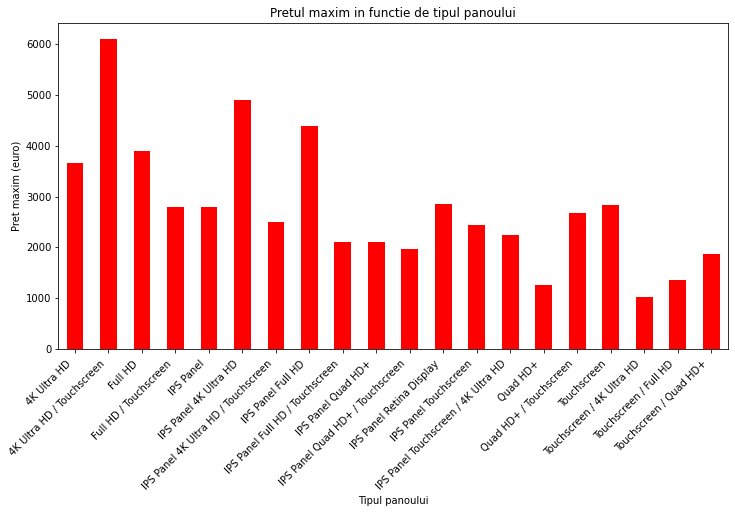
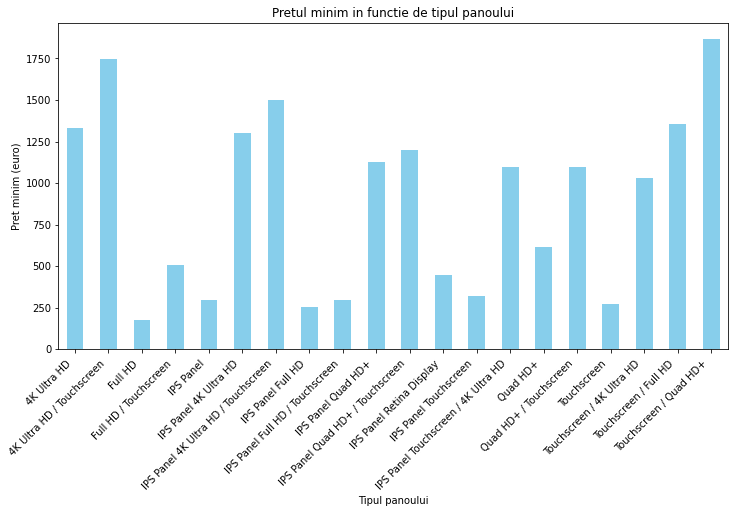
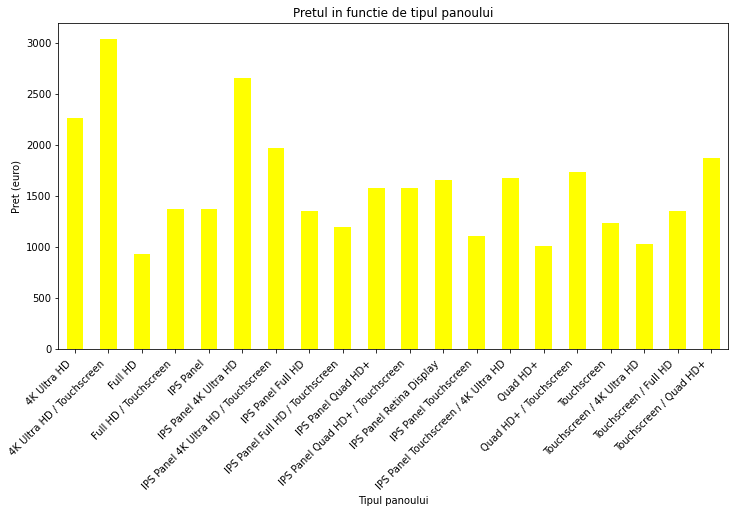
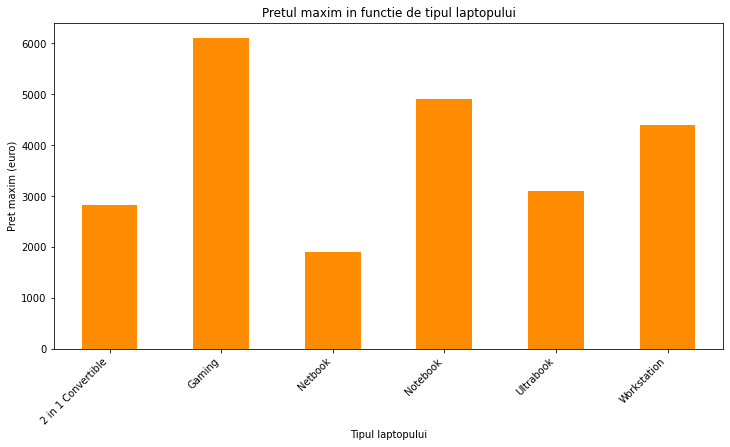
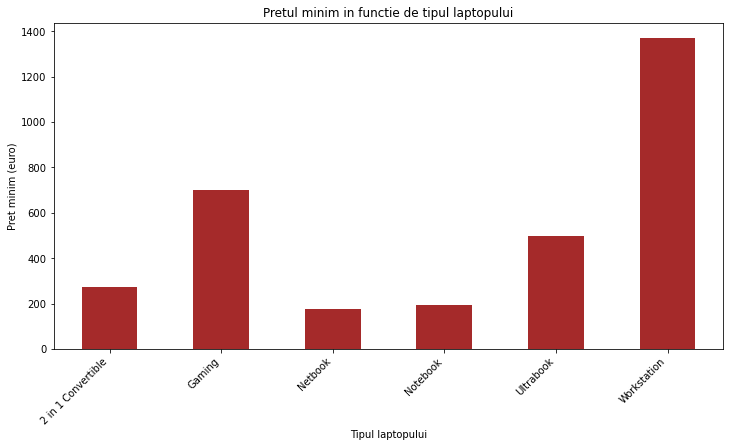
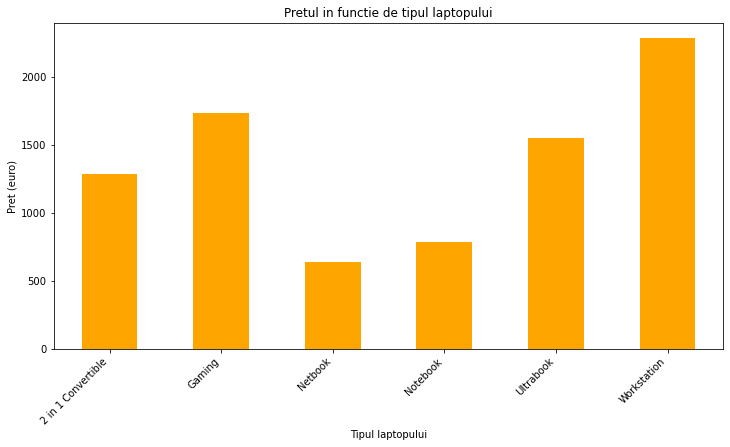
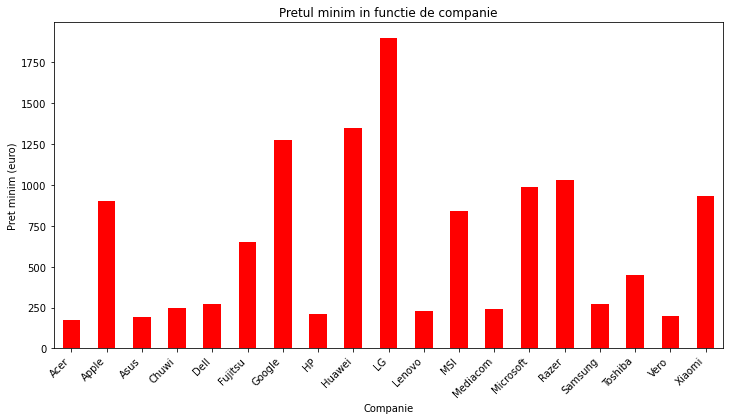
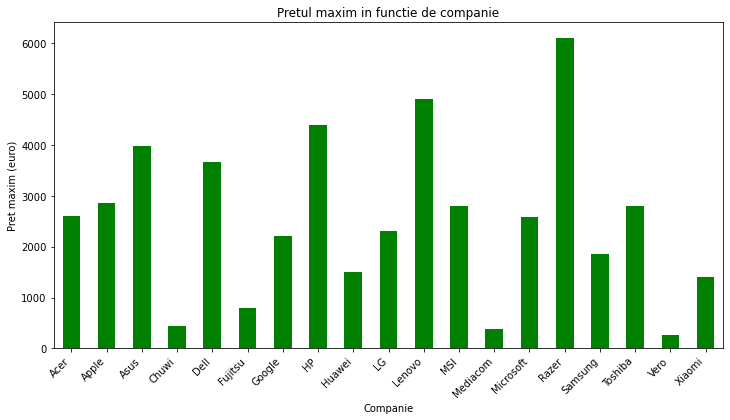
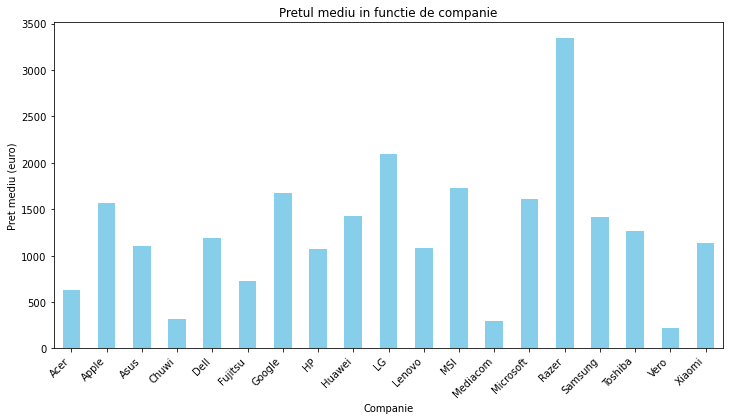
Cel mai nefolosit produs este: Omen 17-W006na

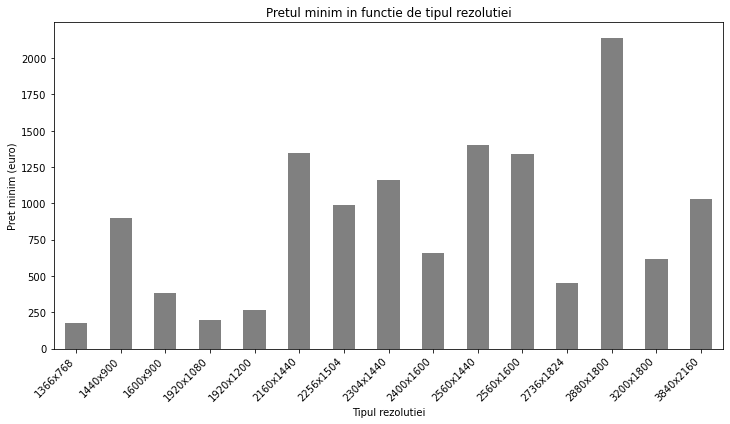
Cea mai nepopulara companie este: Huawei

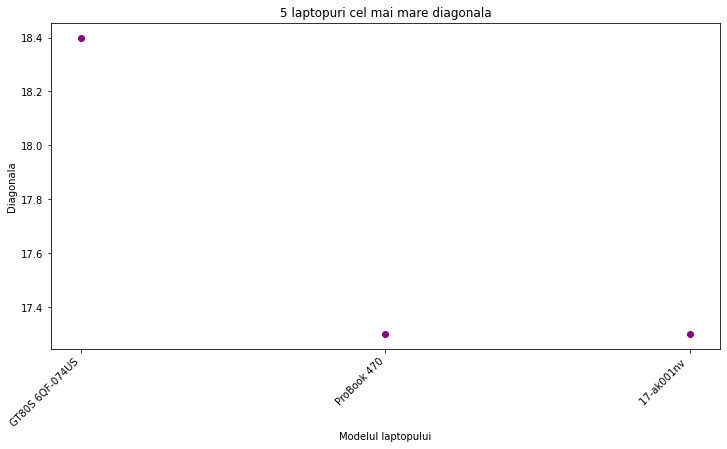
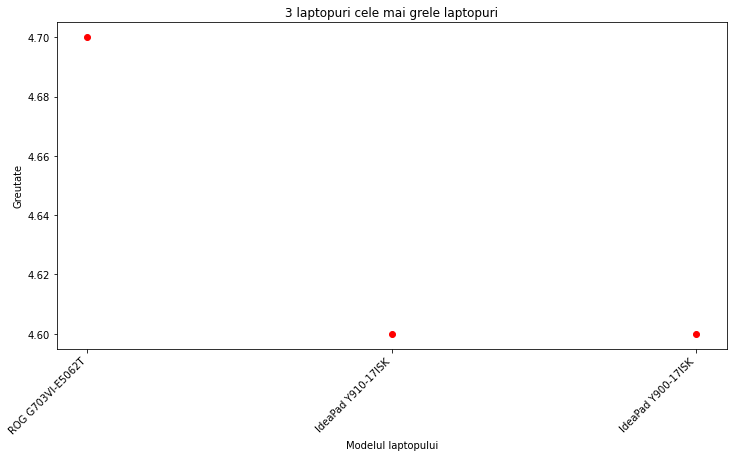
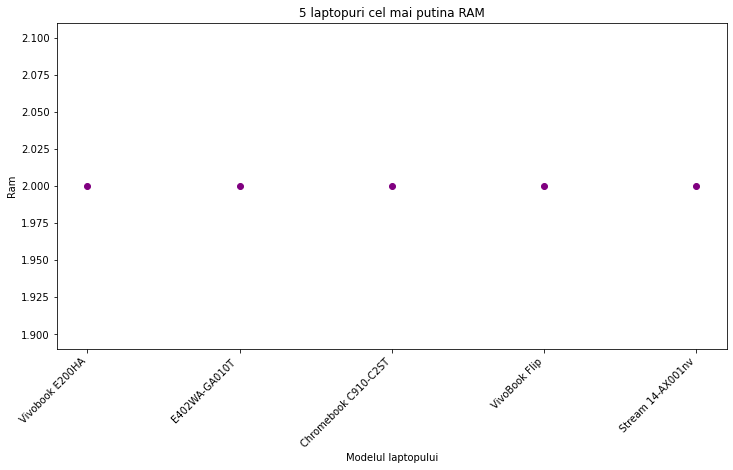
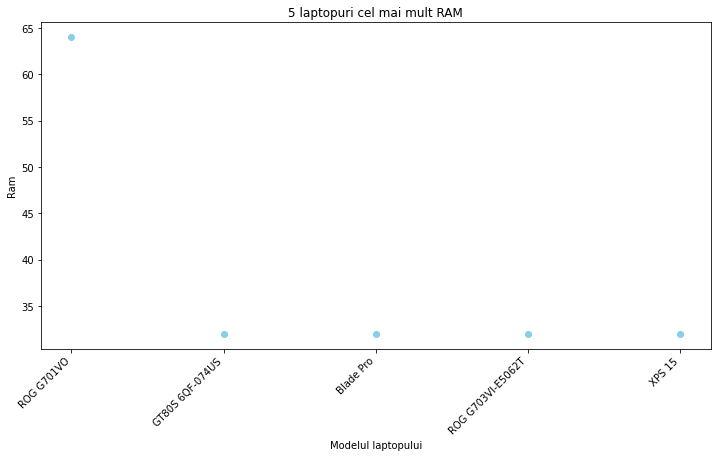
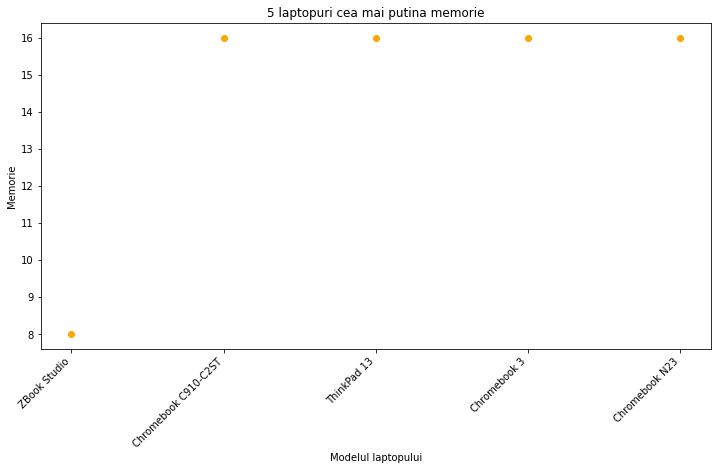
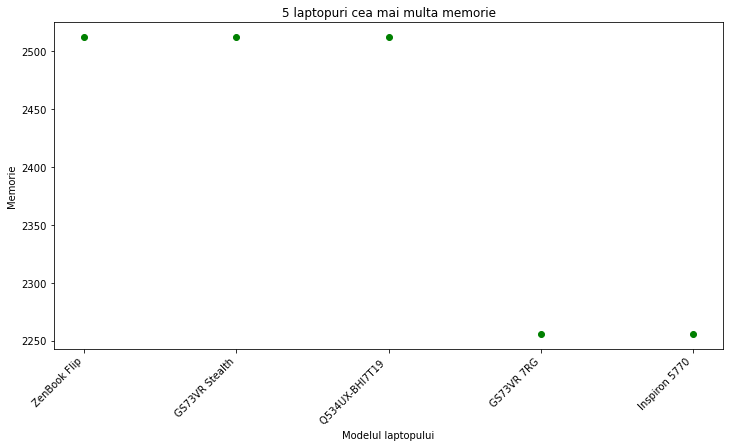
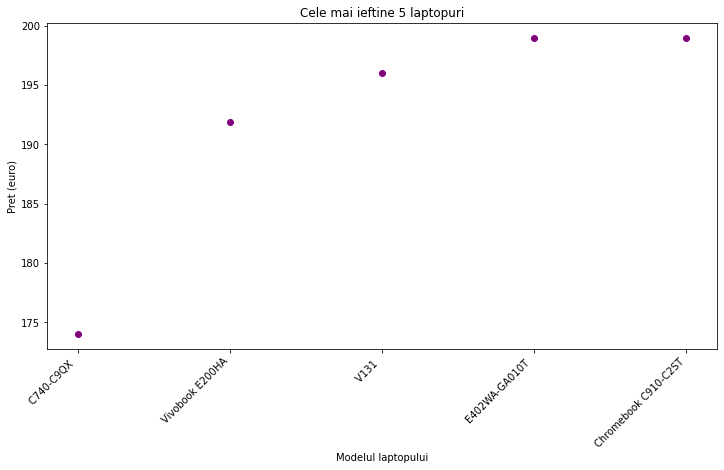
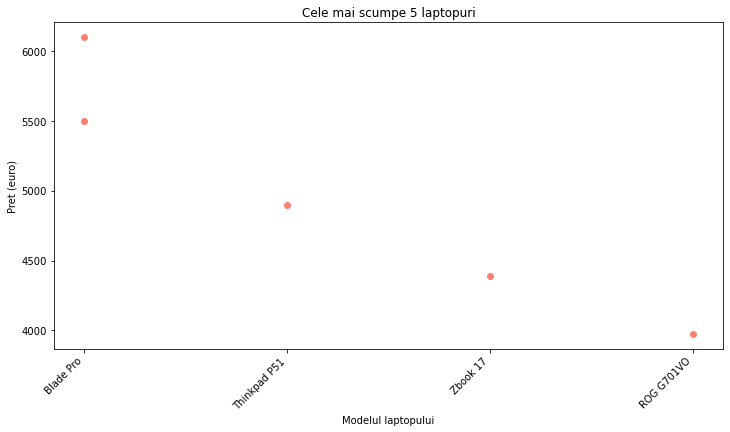
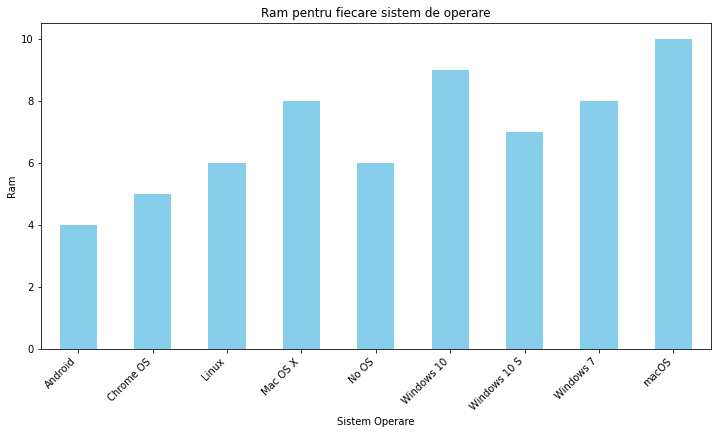
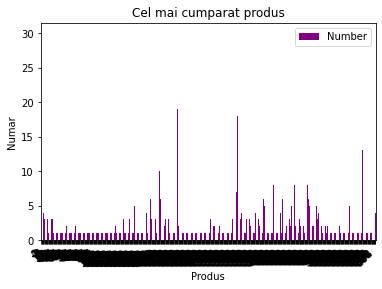
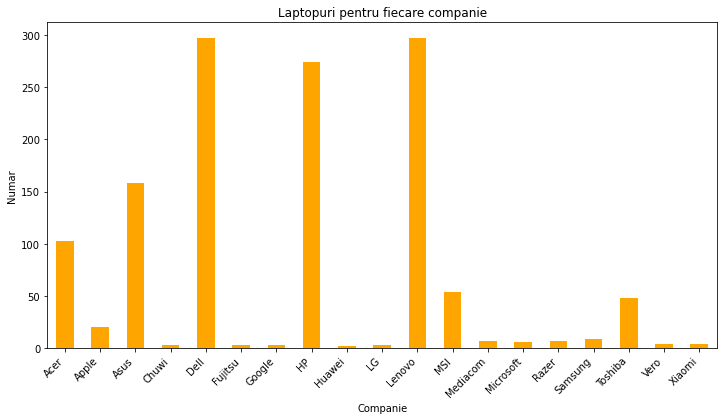
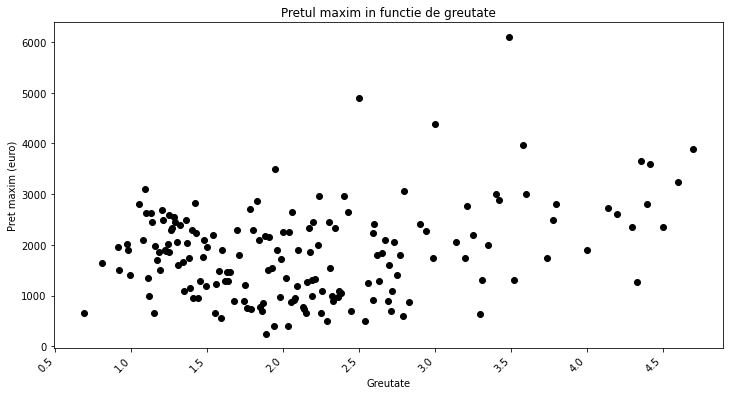
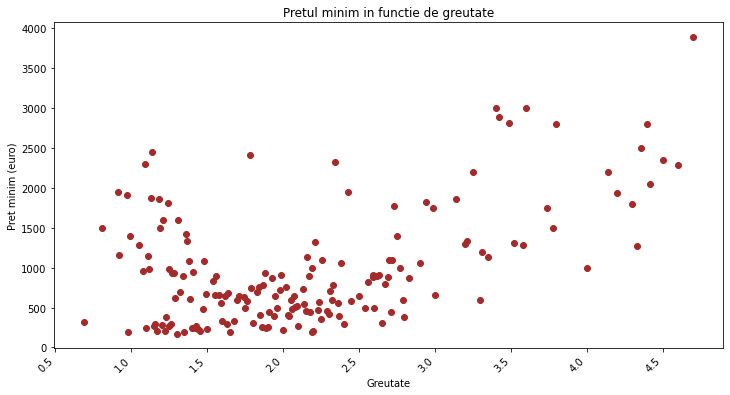
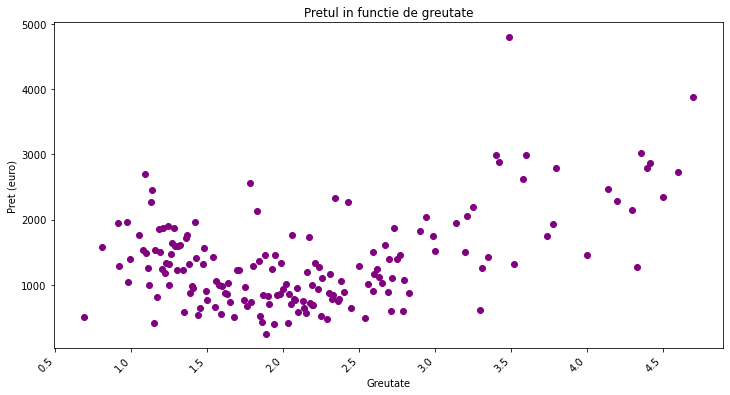
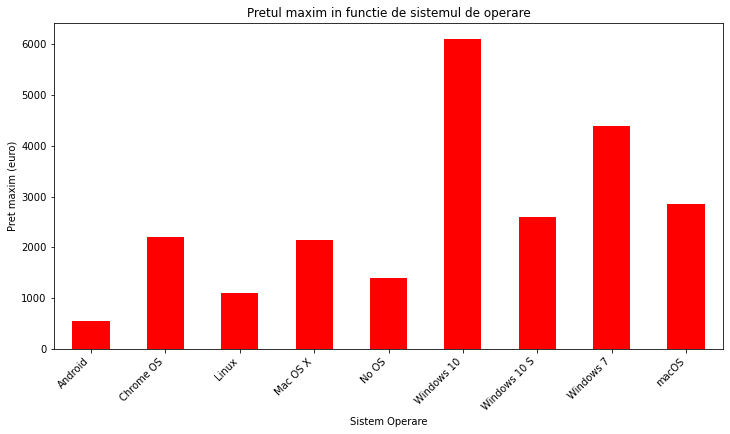
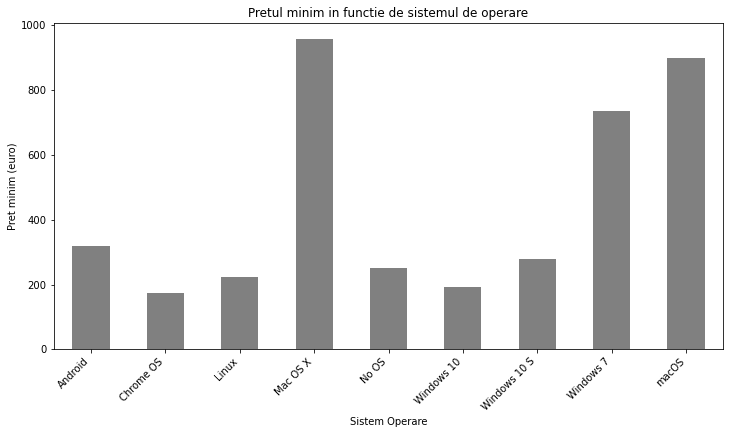
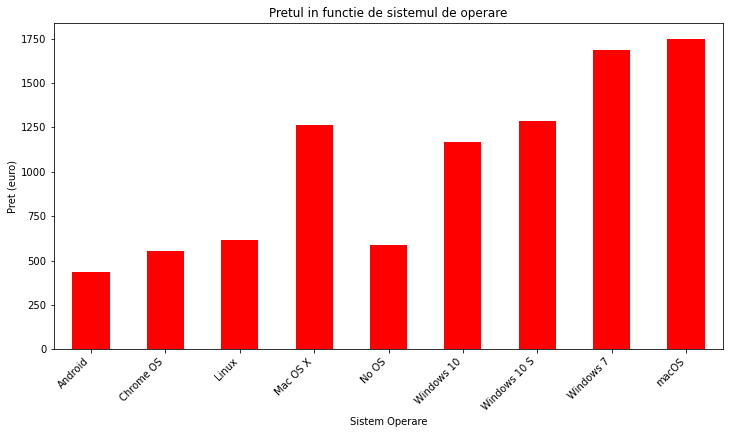
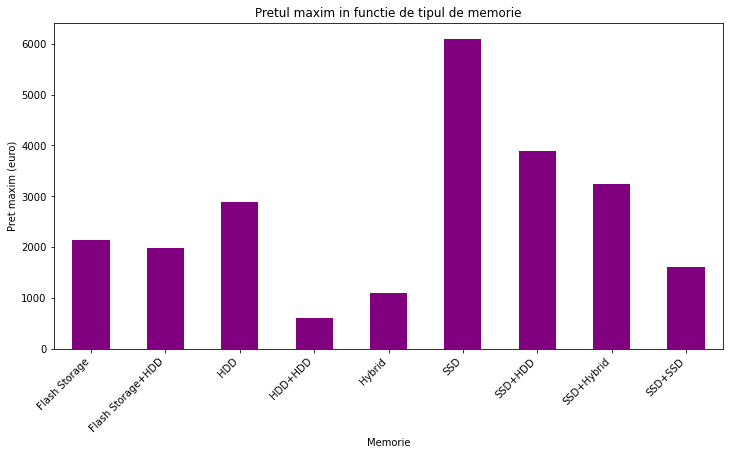
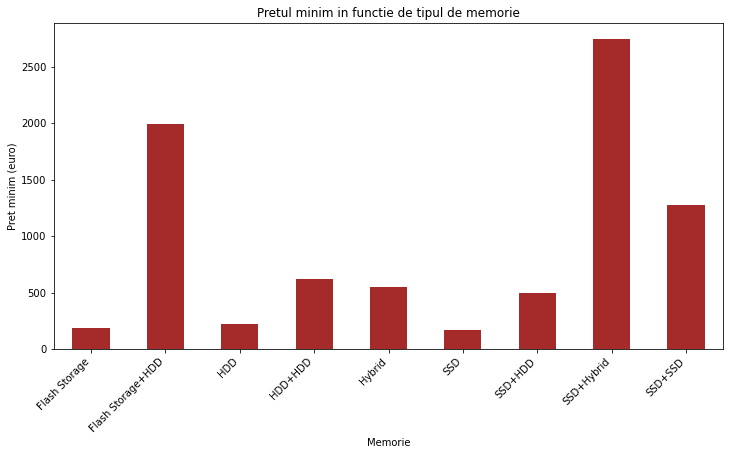
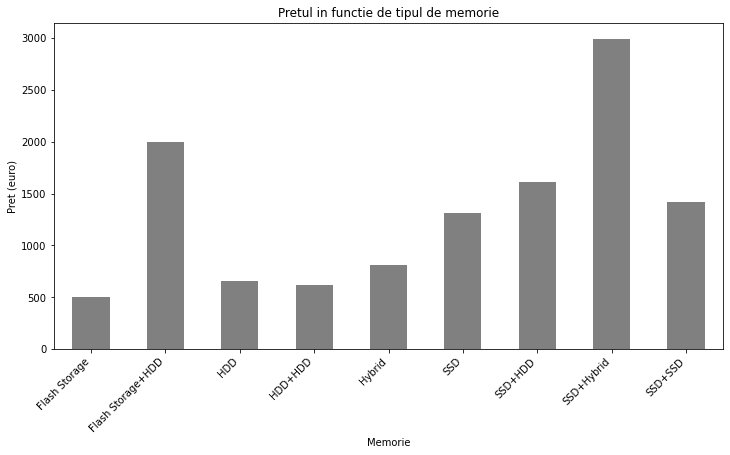
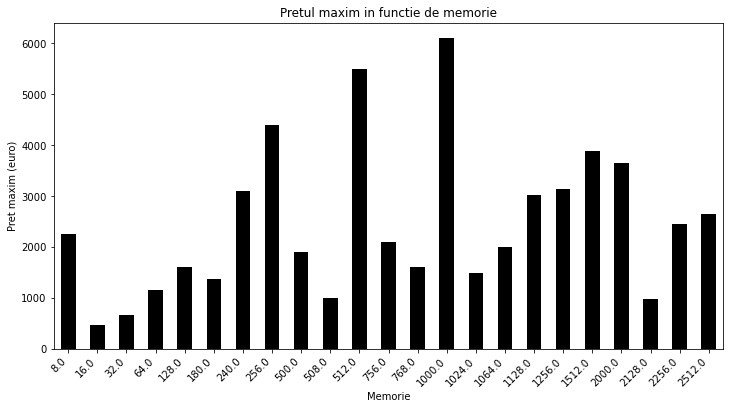
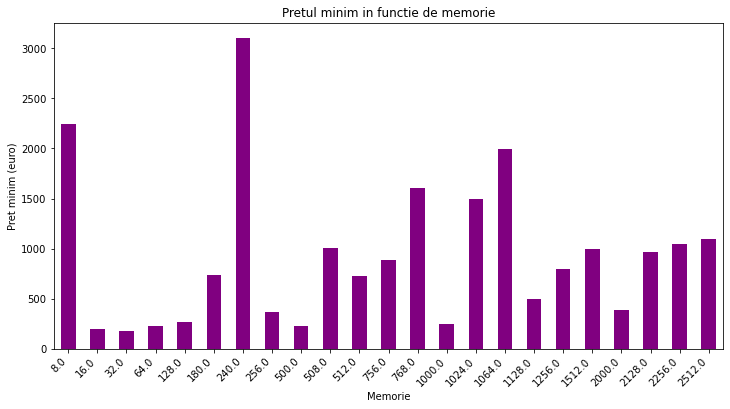
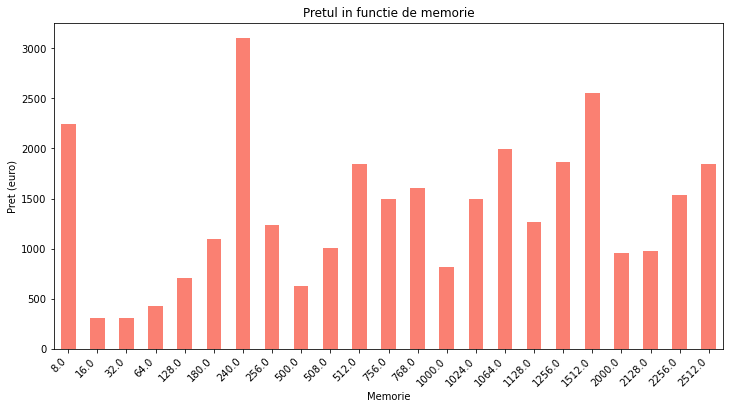
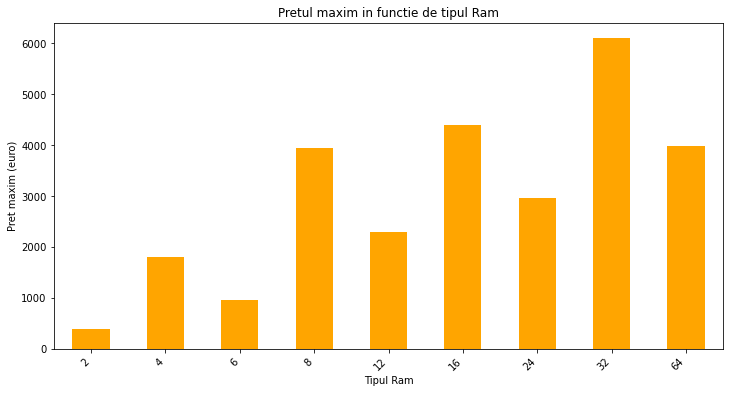
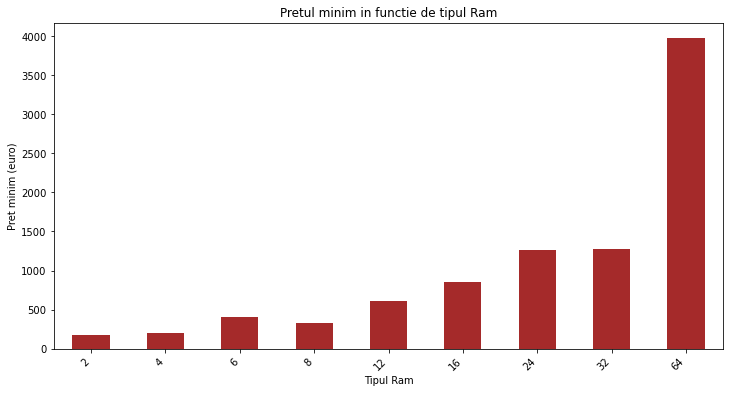
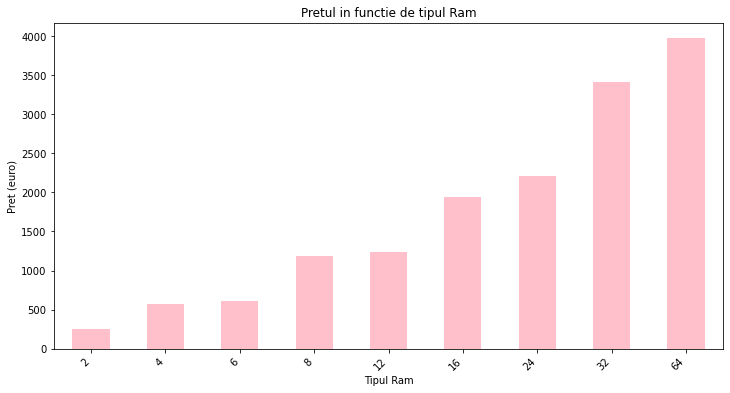
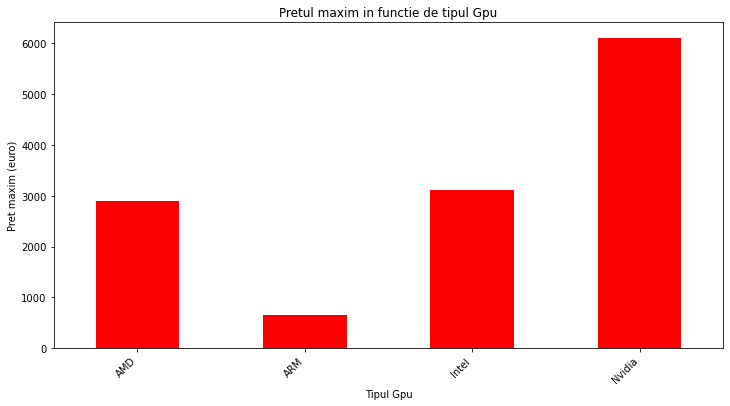
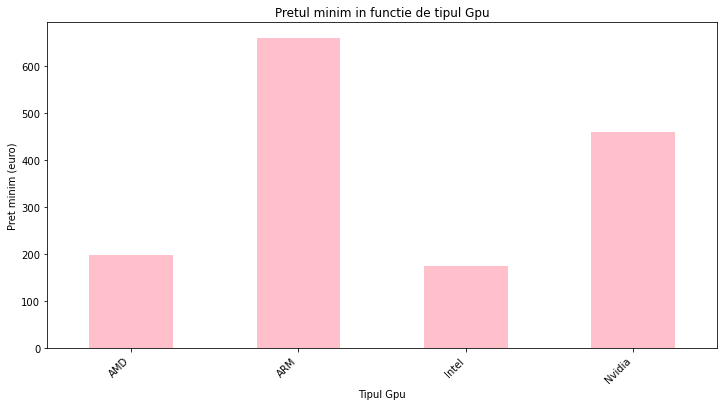
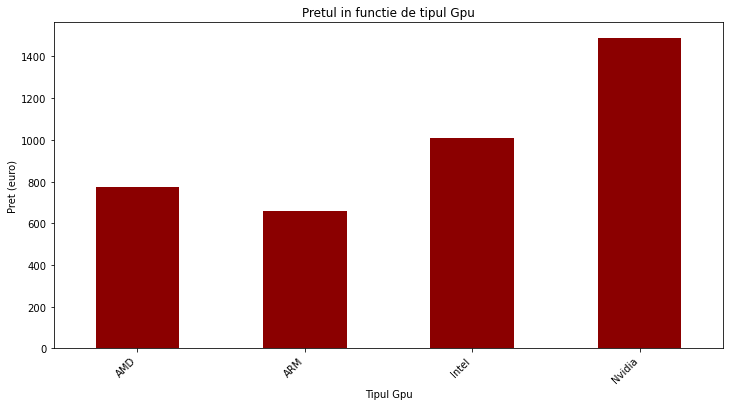
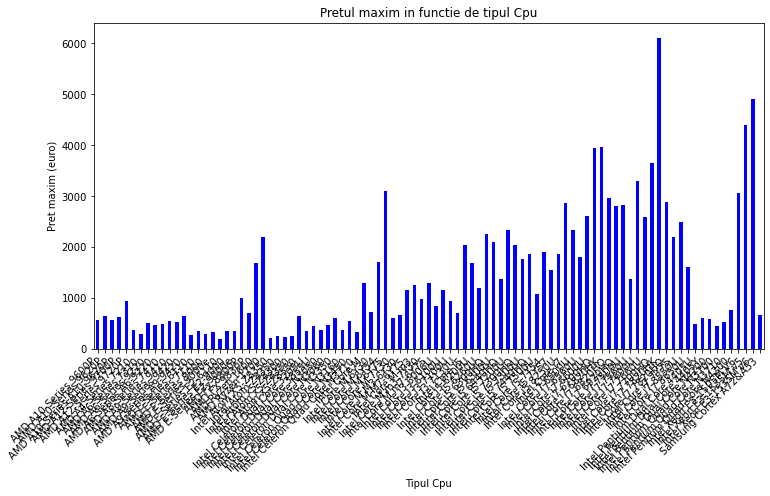
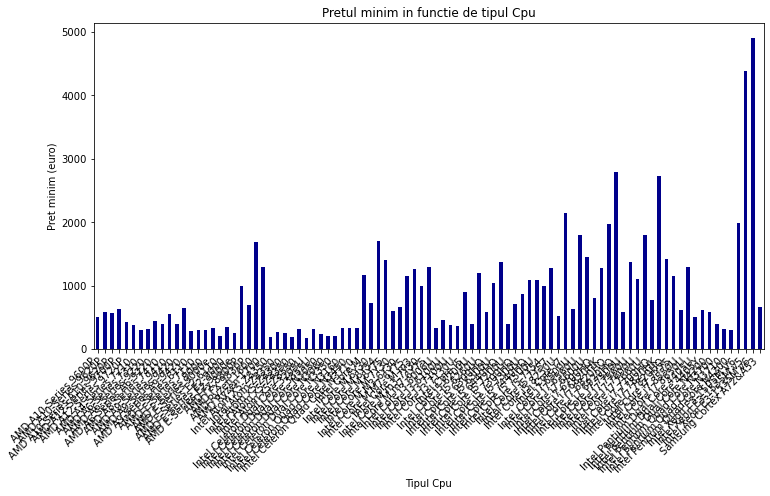
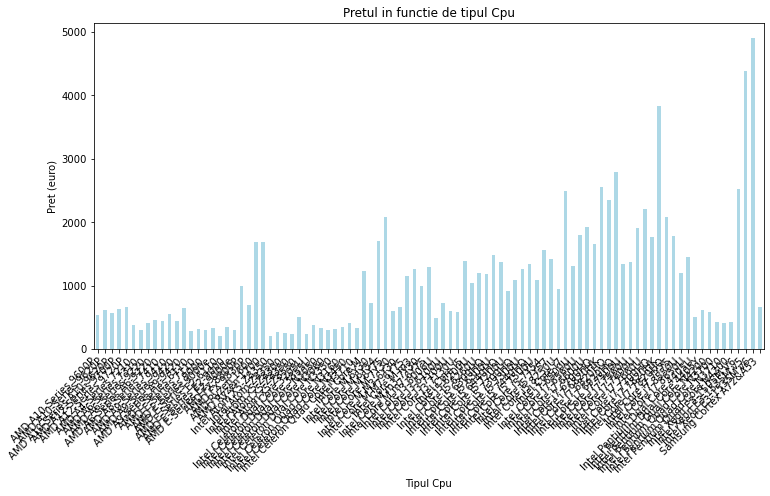
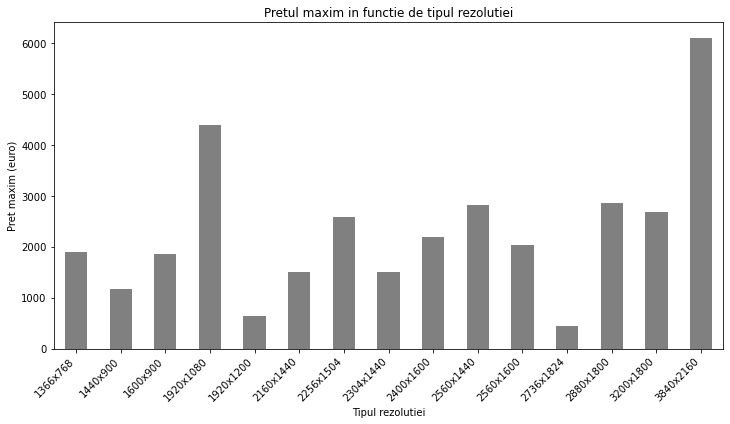
Cel mai nefolosit sistem de operare este: Android

Cea mai nepopulara populara diagonala este: 18.4

In continoare vom atasa doar graficle iar DataFrame-urile le puteti gasi in cod:







Coloanele categoriale alese sunt: “TypeName”, ”OpSys”. Am ales aceste coloane deoarece pot lua valori dintr-un set finit de categorii.

Am sters coloanele care nu sunt numerice sau categoriale

coloane\_categoriale = df.select\_dtypes(include=["object"]).columns.drop(['Company', 'Product', 'Cpu', 'Gpu', 'Memory', 'PanelType', 'Resolution', 'Gpu\_Model'])

Transformam coloanele categoriale in dummies

df\_encoded = pd.get\_dummies(df, columns=coloane\_categoriale, drop\_first=True, dtype=int)

Cand facem regresie este important sa folosim variabile dummy pentru a trata aceste coloane categoriale.

Selecteaza caracteristicile pentru regresie(variabile independente)

features = df\_encoded.drop(['Company', 'Product', 'Cpu', 'Gpu', 'Memory', 'PanelType', 'Resolution', 'Gpu\_Model','Price\_euros'], axis=1)

Selecteaza variabila tinta(variabila dependenta)

target = df\_encoded['Price\_euros']

Variabila cea mai importanta pentru regresie din setul de date este “Price\_euroes” intrucat regresia se face pentru a prezice pretul din viitor al unui anumit tip de laptop.

Dupa ce facem regresia vom vedea ca acestea sunt cele mai importante variabile deoarece au p values mai mici de 0,05

print(ols\_model.summary())

Rezultat:

OLS Regression Results

==============================================================================

Dep. Variable: Price\_euros R-squared: 0.726

Model: OLS Adj. R-squared: 0.721

Method: Least Squares F-statistic: 135.5

Date: Fri, 26 Jan 2024 Prob (F-statistic): 1.99e-270

Time: 16:05:36 Log-Likelihood: -7595.9

No. Observations: 1042 AIC: 1.523e+04

Df Residuals: 1021 BIC: 1.534e+04

Df Model: 20

Covariance Type: nonrobust

========================================================================================

coef std err t P>|t| [0.025 0.975]

----------------------------------------------------------------------------------------

const 197.2985 300.880 0.656 0.512 -393.115 787.712

laptop\_ID 0.0713 0.031 2.287 0.022 0.010 0.133

Inches -43.8968 17.425 -2.519 0.012 -78.090 -9.704

GHz 189.1737 25.678 7.367 0.000 138.786 239.562

RamGB 76.9392 2.873 26.780 0.000 71.301 82.577

WeightKG 95.3037 38.020 2.507 0.012 20.698 169.909

MemoryGB -0.0969 0.023 -4.266 0.000 -0.142 -0.052

MemoryBonus 0.0698 0.025 2.813 0.005 0.021 0.118

MemoryTotal -0.0272 0.020 -1.338 0.181 -0.067 0.013

TypeName\_Gaming -136.0491 62.349 -2.182 0.029 -258.396 -13.702

TypeName\_Netbook -355.6568 96.329 -3.692 0.000 -544.682 -166.631

TypeName\_Notebook -307.2866 45.506 -6.753 0.000 -396.583 -217.990

TypeName\_Ultrabook 133.0808 48.865 2.723 0.007 37.193 228.969

TypeName\_Workstation 632.9771 89.139 7.101 0.000 458.060 807.894

OpSys\_Chrome OS 253.2581 269.617 0.939 0.348 -275.809 782.325

OpSys\_Linux 306.1988 265.202 1.155 0.249 -214.205 826.602

OpSys\_Mac OS X 489.8079 305.935 1.601 0.110 -110.525 1090.140

OpSys\_No OS 278.4786 265.705 1.048 0.295 -242.912 799.869

OpSys\_Windows 10 477.7300 260.086 1.837 0.067 -32.635 988.095

OpSys\_Windows 10 S 540.1516 293.520 1.840 0.066 -35.820 1116.123

OpSys\_Windows 7 802.3404 267.225 3.002 0.003 277.968 1326.713

OpSys\_macOS 801.0446 288.342 2.778 0.006 235.234 1366.855

==============================================================================

Omnibus: 370.314 Durbin-Watson: 1.965

Prob(Omnibus): 0.000 Jarque-Bera (JB): 3010.310

Skew: 1.403 Prob(JB): 0.00

Kurtosis: 10.840 Cond. No. 2.55e+15

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 2.02e-22. This might indicate that there are

strong multicollinearity problems or that the design matrix is singular.

Corelatie df

correlation\_matrix = df.corr(numeric\_only=True)

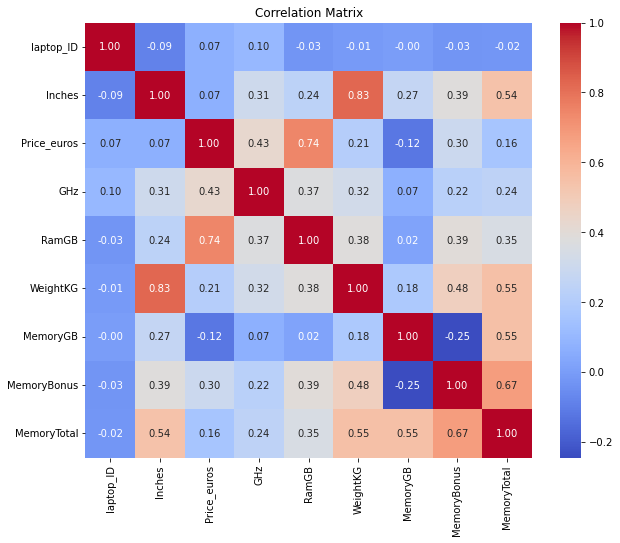
plt.figure(figsize=(10, 8))

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Matrix')

plt.show()

Rezultat:



Corelatie df\_encoded

correlation\_matrix\_encoded = df\_encoded.corr(numeric\_only=True)

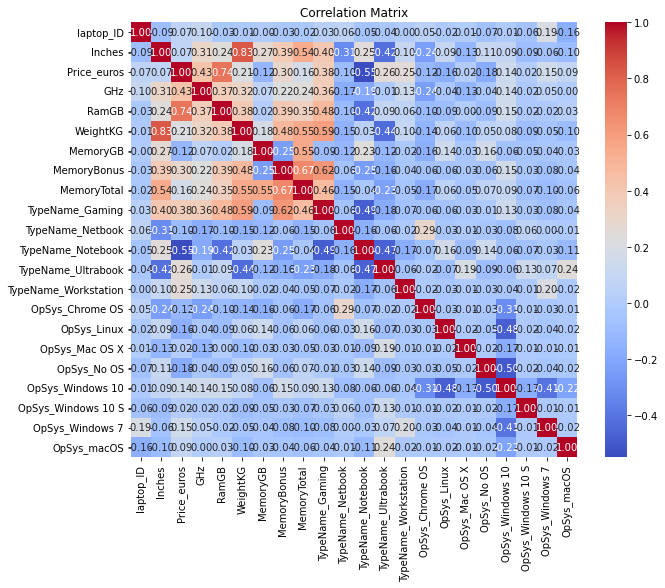
plt.figure(figsize=(10, 8))

sns.heatmap(correlation\_matrix\_encoded, annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Matrix')

plt.show()

Rezultat:



1: Corelatie perfecta pozitiva - variabilele cresc impreuna.

0: Lipsa de corelatie - variabilele nu sunt asociate.

-1: Corelatie perfecta negativa - variabilele scad impreuna.

Corelatie pozitiva, adica cresterea unei variabile este asociata cu cresterea celeilalte variabile.

Corelatie negativa, adica cresterea unei variabile este asociata cu scaderea celeilalte variabile.

Cu cat coeficientul de corelatie este mai aproape de 1 sau -1, cu atat relatia este mai puternica.

Cu cat coeficientul se apropie de 0, cu atat relatia este mai slaba.

Aproape de 1: Un coeficient aproape de 1 indica o corelatie pozitiva puternica. Acest lucru inseamna ca pe masura ce o variabila creste, si cealalta variabila are tendinta de a creste.

Aproape de -1: Un coeficient aproape de -1 indica o corelatie negativa puternica. Acest lucru inseamna ca pe masura ce o variabila creste, cealalta variabila are tendinta de a scadea.

Aproape de 0: Un coeficient aproape de 0 indica o corelatie liniara slaba sau inexistenta. Variabilele pot avea inca vreo alta forma de relatie, dar nu este liniara.

Pozitiv vs. Negativ: Semnul coeficientului de corelatie (+ sau -) indica directia relatiei. Pozitiv inseamna ca variabilele se misca in aceeasi directie, in timp ce negativ inseamna ca se misca in directii opuse.

Magnitudine: Cu cat coeficientul de corelatie este mai aproape de 1 sau -1, cu atat este mai puternica corelatia. Cu cat este mai aproape de 0, cu atat este mai slaba corelatia.

Exemplu: Daca ne uitam in Matricea de corelatie pentru df putem observa o corelatie puternica intre “Inches” si “WeightKG” avand o valoare de 0.83.

Aceasta arata ca este o corelatie pozitiva si puternica. Fiind pozitiva variabilele cresc impreuna. Acest lucru arata ca variabilele sunt dependente una de cealalta in mare masura. Atunci cand ecranul are o marime mai mare laptop-ul are si o greutate mai mare.

Putem observa o corelatie foarte slaba intre “MemoryGB” si “GHz” avand valoarea de 0,02.

Aceasta arata ca este o corelatie pozitiva foarte slaba. Fiind pozitiva variabilele cresc impreuna.

Acest lucru arata ca variabilele sunt independente una de cealalta. Numarul de GB din discurile de memorie nu influenteaza GHz procesorului.

Putem observa o corelatie slaba intre “MemoryGB” si “MemoryBonus” avand o valoare de -0.25.

Acest lucru arata ca este o corelatie negativa si slaba. Fiind negativa variabilele se misca in directii opuse. Cand “MemoryGB” creste “MemoryBonus” scade intr-o oarecare masura.

Acest lucru arata ca variabilele sunt dependente una de cealalta intr-o mica masura. Cand capacitatea de memorie avuta este mare atunci este mai putin probabil sa mai avem nevoie de memorie extra.

Un alt exemplu este cel dintre “Price\_euroes” si “RamGB” avand o valoare de 0.74.

Aceasta arata ca este o corelatie pozitiva si puternica. Fiind pozitiva variabilele cresc impreuna. Acest lucru arata ca variabilele sunt dependente una de cealalta in mare masura.

Cand creste capacitatea Ram a unui laptop ii creste si pretul acestuia.

Asa putem face pentru toate variabilele din matricea de corelatii.

Coeficientul lui Pearson

correlationresults = []

#Calculați și adăugați coeficientul de corelație Pearson în listă

for feature in features.columns:

correlation, = pearsonr(features[feature], target)

correlation\_results.append({'Feature': feature, 'Pearson Correlation': correlation})

#Creare DataFrame din lista de rezultate

correlation\_df = pd.DataFrame(correlation\_results)

#Afișare DataFrame cu rezultatele

print(correlation\_df)

1 indica o corelatie pozitiva perfecta (inseamna ca o crestere intr-o variabila este asociata cu o crestere proportionala in cealalta variabila).

-1 indica o corelatie negativa perfecta (inseamna ca o crestere intr-o variabila este asociata cu o scadere proportionala in cealalta variabila).

0 indica lipsa unei corelatii lineare intre variabile.

In general, interpretati coeficientul astfel:

0-0.3: Corelatie slaba.

0.3-0.7: Corelatie moderata.

0.7-1.0: Corelatie puternica.

Regresie

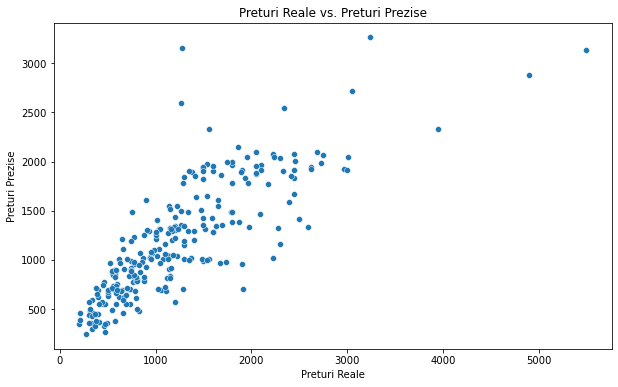
Un coeficient de determinare R^2 apropiat de 1 indica o potrivire buna a modelului cu datele analizate.

1. Regresia liniara

In acest script Python, se foloseste regresia liniara in scopul predictiei pretului unui laptop pe baza unui set de caracteristici. Regresia liniara este alesa pentru ca se doreste sa se evalueze relatia liniara dintre variabilele de intrare si variabila de iesire (pretul).

Regresia liniara ofera o interpretare simpla a relatiei dintre variabile. Coeficientii asociate variabilelor de intrare indica modificarea medie a variabilei de iesire pentru o modificare unitara in variabila de intrare respectiva.

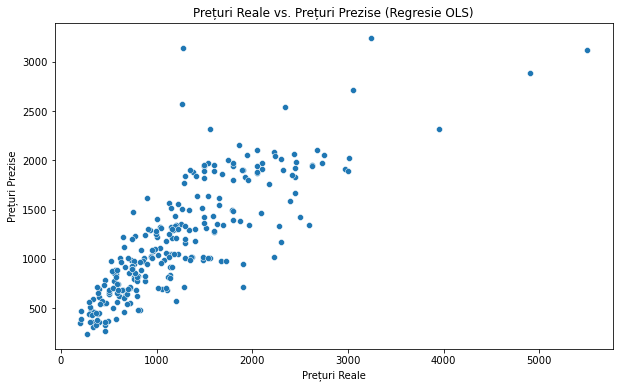
Ea se bazeaza pe asumari clare, precum liniaritatea relatiei si independenta erorilor, ceea ce face modelul mai interpretabil si usor de inteles.



R^2(scorul\_r2)=0.658

1. OLS-model

Scopul utilizării regresiei liniare OLS în setul de date este de a dezvolta un model predictiv care poate să estimeze prețul unui laptop pe baza altor caracteristici disponibile în setul de date. Regresia liniară este o metodă simplă, dar puternică, în special atunci când există o relație liniară între variabilele independente și variabila dependenta

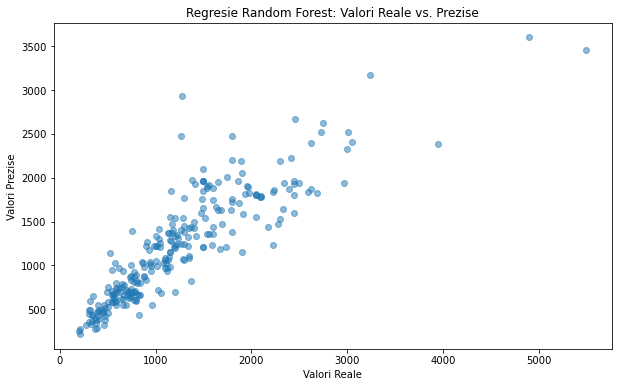


R^2(r2rOLS)=0.659

1. Random-Forest

Random Forest poate sa detecteze si sa modeleze relatii non-liniare intre caracteristici si variabila tinta. Acest lucru este util atunci cand datele au relatii complexe si nu pot fi modelate eficient cu o regresie liniara simpla.

Random Forest furnizeaza o masura a importantei caracteristicilor, ceea ce poate fi util pentru a identifica caracteristicile care contribuie cel mai mult la predictia preturilor laptopurilor.



R^2(r2)= 0.772

Multicoliniaritatea

De asemenea, este demn de mentionat ca in analiza datelor sunt utilizate si alte tehnici precum analiza de importanta a caracteristicilor, VIF (factori de inflatie a variantei) pentru a evalua multicoliniaritatea si Recursive Feature Elimination (RFE) pentru selectia caracteristicilor. Aceste tehnici pot aduce o perspectiva suplimentara si pot contribui la imbunatatirea modelului de predictie.

Determinam importanta coloanelor viitoare selectand coeficientii si p values

rez\_regresie = pd.DataFrame({

"Coeficient": coeficienti,

"p-values": p\_values

})

r\_squared = ols\_model.rsquared

R^2=0.726

Rezulta ca valorile din DataFrame sunt importante

Factorii de inflație a varianței (VIF) sunt utilizați pentru a evalua gradul de coliniaritate între variabilele independente dintr-un model de regresie. Coliniaritatea între variabilele independente poate afecta performanța modelului, făcând interpretarea coeficienților mai dificilă și conducând la variabilitate crescută a acestora. Dacă există coliniaritate semnificativă, VIF-urile cresc, semnalând probleme în interpretarea efectelor individuale ale variabilelor independente.

VIF < 5: Indica o multicolinearitate scazuta. Variabilele nu sunt foarte corelate.

5 < VIF < 10: Indica o multicolinearitate moderata. Este un semnal ca poate exista o anumita corelatie, dar nu este severa.

VIF > 10: Indica o multicolinearitate ridicata. Acest lucru sugereaza un nivel problematic de corelatie intre variabile.

In ceea ce priveste legatura cu regresia, atunci cand utilizam VIF pentru a evalua coliniaritatea intr-un model de regresie liniara multipla, obiectivul este sa identificam și sa eliminam variabilele predictor care sunt coliniare. O valoare mare a VIF-ului pentru o anumita variabila indica faptul ca acea variabila poate fi prevăzuta destul de bine de celelalte variabile din model, ceea ce poate duce la probleme in interpretarea coeficientilor si in evaluarea impactului fiecarei variabile asupra variabilei raspuns.

Variable VIF

0 const 856.492547

1 laptop\_ID 1.132601

2 Inches 4.666699

3 GHz 1.410524

4 RamGB 1.673323

5 WeightKG 4.890804

6 MemoryGB inf

7 MemoryBonus inf

8 MemoryTotal inf

9 TypeName\_Gaming 4.172379

10 TypeName\_Netbook 1.348014

11 TypeName\_Notebook 4.072143

12 TypeName\_Ultrabook 2.499709

13 TypeName\_Workstation 1.434291

14 OpSys\_Chrome OS 14.612391

15 OpSys\_Linux 31.945695

16 OpSys\_Mac OS X 5.144048

17 OpSys\_No OS 33.855259

18 OpSys\_Windows 10 99.144895

19 OpSys\_Windows 10 S 5.156063

20 OpSys\_Windows 7 23.738835

21 OpSys\_macOS 7.650911  
  
Pentru a optimiza modelul trebuie sa stergem coloanele care au VIF>10

features = df\_encoded.drop(['Company', 'Product', 'Cpu', 'Gpu', 'Memory', 'PanelType', 'Resolution', 'Gpu\_Model','Price\_euros','MemoryGB','MemoryBonus','OpSys\_Linux','OpSys\_Chrome OS','OpSys\_No OS','OpSys\_Windows 10','OpSys\_Windows 7'], axis=1)

Rezultat:  
 Variable VIF

0 const 321.377655

1 laptop\_ID 1.084599

2 Inches 4.630714

3 GHz 1.363745

4 RamGB 1.652930

5 WeightKG 4.858077

6 MemoryTotal 1.637981

7 TypeName\_Gaming 3.786236

8 TypeName\_Netbook 1.292662

9 TypeName\_Notebook 4.020534

10 TypeName\_Ultrabook 2.451259

11 TypeName\_Workstation 1.378137

12 OpSys\_Mac OS X 1.068293

13 OpSys\_Windows 10 S 1.032518

14 OpSys\_macOS 1.094910

Acest lucru determina cresterea acuratetii modelului si scaderea multicoliniaritatii.

Recursive Feature Elimination (RFE) este o tehnică de selecție a caracteristicilor care implică eliminarea iterativă a celor mai puțin semnificative caracteristici din setul de date. Aceasta este adesea utilizată în contextul regresiei pentru a îmbunătăți performanța modelului și pentru a simplifica interpretarea acestuia.

RFE ne ajută să identificam și să păstram doar cele mai relevante caracteristici pentru model, eliminând treptat cele mai puțin informative. Acest lucru poate fi util pentru a reduce complexitatea modelului și pentru a îmbunătăți eficiența în timpul antrenării și evaluării.

rfe = RFE(estimator = model, n\_features\_to\_select=10)

rf = rfe.fit(features, target)

atributeselectate = pd.DataFrame({

"atribut": features.columns,

"selectate": rfe.support,

"ranking": rfe.ranking\_

})

Pregatim setul pentru regresiile optimizate

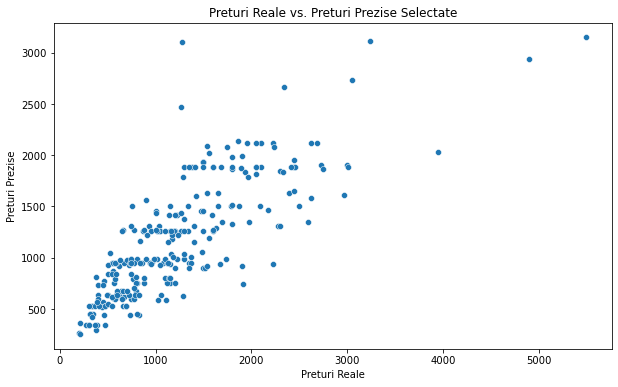
#Regresie pe modelul optimizat

atribute\_selectate\_pt\_reg = atribute\_selectate[atribute\_selectate["selectate"]]["atribut"]

features\_selected= features2[atribute\_selectate\_pt\_reg]

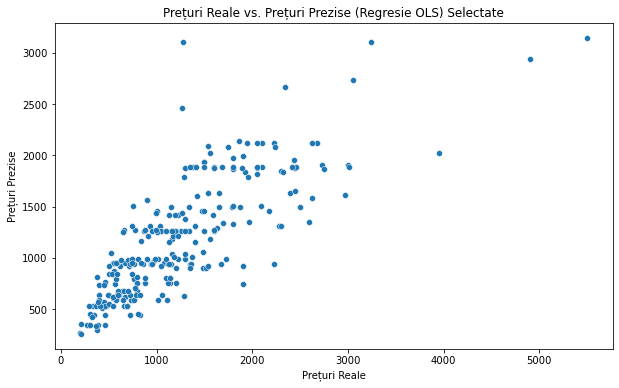
x\_tr\_sel, x\_test\_sel, y\_tr\_sel, y\_test\_sel = train\_test\_split(features\_selected, target, test\_size = 0.2, random\_state=0)

1. Regresia liniara optimizata:



R^2(r2\_scor\_selectat):0,632

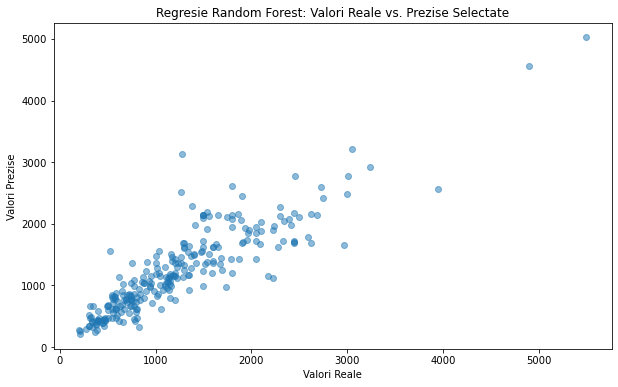
1. Regresia OLS optimizata:



R^2(r2rOLS\_selectat):0,6598

Aici putem observa ca este acelasi R^2 deoarece atributele pentru regresie au fost optimizate la inceput.

1. Regresia Random Forest optimizata:



R^2(r2\_select):0,781

Aici s-a imbunatatit R^2 deoarece am scapat de multicoliniaritate si am utilizat atributele optimizate.

Exemplu de interpretare a regresiei:Folosind coeficienti aflati la regresia cu model OLS din OLS Sumary:  
Laptop cu 2,3 GHz, 8GB memorie ram,greutate de 1,37,memorie interna de 128gb,Ultrabook cu MacOS X,ecran de 13.3 inches.

Calcul: 2,3\*189,1737+8\*76,9392+1,37\*95,3037+128\*(0,969)+133,0808\*1+13,3\*(-43,8968)+1\*489,8079=1096,20 euros