3.2.2. Predicţia punctuală

Pe baza modelului de regresie liniară aplicat se poate face o previziune punctuală a comportamentului variabilei y, în funcţie de valorile fixe pe care le ia variabila independentă x. Rezultatele obţinute se referă la un comportament mediu al variabilei y [19].

3.2.3. Estimarea parametrilor pe bază de interval de încredere

Predicţia punctuală a unui parametru poate fi la o anumită distanţă faţă de valoarea reală a parametrului estimat. Prin urmare estimarea parametrului se poate face şi pe baza unui interval de predicţie, între limitele căruia poate sa ia valori parametrul estimat, în cazul nostru variabila estimată ŷ.

Estimarea pe interval de predicţie se face pe baza unui **interval de încredere** în cadrul căruia se găseşte cu probabilitate mare parametrul estimat, modul de calcul fiind de forma: estimare ± eroare, astfel că estimatorul să se situeze între valoarea inferioară a intervalului şi valoarea superioară a acestuia.

[valoarea inferioară < estimator < valoarea superioară] = 1-α,

unde α este nivelul de semnificaţie, iar intervalul definit de cele două valori, cea inferioară şi cea superioară va cuprinde estimatorul populaţiei cu o probabilitate de 1-α [19].

$$[\hat{y} - t_{1-\frac{\alpha}{2}} * \sigma_{\hat{y}} ; \hat{y} + t_{1-\frac{\alpha}{2}} * \sigma_{\hat{y}}]$$

Calculul intervalului de predicţie se face pe baza următorilor indicatori [19]:

- variaţia erorilor de predicţie/reziduurilor, care se calculează ca şi rădăcina pătratului diferenţelor între valorile lui y şi valorile estimate a lui y prin aplicarea modelului (după legea Student s-au luat n-2 grade de libertate)

$$\sigma_{\varepsilon} = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{n-2}}$$

- variaţia estimatorului ŷ calculat în funcţie de x după formula:

$$\sigma_{\hat{y}} = \sigma_{\varepsilon} \sqrt{1 + \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum(x_i - \bar{x})^2}}$$

- gradul de încredere al intervalului specificat – scorul z, care corespunde nivelului de încredere; valoarea critică z se găseşte din tabelul de scoruri z, întrucât se cunoaşte valoarea abaterii standard a populaţiei şi se poate presupune că valorile sunt distribuite

normal; (valorile z comune sunt 1,645 pentru un nivel de încredere de 90%, 1,960 pentru un nivel de încredere de 95% si 2,576 pentru un nivel de încredere de 99%).

În funcţie de indicatorii prezentaţi se calculează **intervalul de predicţie** cu un nivel de încredere de (1-α) [19] după urmatoarea formulă:

$$\hat{y}_i \pm t_{1-\frac{\alpha}{2}} * \sigma_\varepsilon \sqrt{1 + \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum (x_i - \bar{x})^2}}$$

Spre deosebire de intervalul de predicţie, **intervalul de încredere** pentru valorile estimate pe baza modelului, se calculează după formula [19]:

$$\hat{y}_i \pm t_{1-\frac{\alpha}{2}} * \sigma_\varepsilon \sqrt{\frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum (x_i - \bar{x})^2}}$$

3.3. Regresia RANSAC (random simple consensus)

Întrucat regresia liniară este un model conceput pentru a se adapta la toate punctele care reprezintă setul de date, este puternic influenţată de elementele extreme prezente în setul de date (erori de măsurare), astfel că acestea trebuiesc identificate şi eliminate prin metode statistice de analiză a datelor. Metoda celor mai mici pătrate este un model adecvat dacă ipotezele de la care porneşte analiza sunt adevărate şi setul de date nu este afectat de valori extreme (observaţii care nu urmează tiparul observaţiilor majoritare din setul de date).

Statistica robustă (solidă) este implementată pentru a depăşi limitările metodelor tradiţionale de analiză statistică, regresia robustă fiind o metodă statistică concepută pentru modelări care să nu fie afectate de valori extreme sau aberante [24].

Estimatorul RANSAC este un model statistic robust, care rămane relevant în reprezentarea unui set de date, întrucât nu este influenţat de anomalii (outliers), deoarece utilizează în modelarea liniară doar elementele din mulţimea de consens (inliers), care sunt determinate în mod iterativ prin eşantionări aleatorii de date. RANSAC este o metodă iterativă de detectare şi de eliminare a anomaliilor dintr-un set de date, care nu încearcă să adapteze elementele extreme, ci să elimine valorile care nu sunt distribuite normal [25].

Pachetul sklearn pentru Python facilitează implementarea estimatorului RANSAC prin RANSACRegressor, care are ca estimator de bază pentru antrenarea datelor regresia liniară. RANSACRegressor utilizează abaterea absolută mediană (MAD) a variabilei dependente *y* pentru detectarea anomaliilor, întrucât faţă de medie, mediana este un indicator statistic robust [26]. Abaterea absolută mediană este un indicator al deviaţiei de la valoarea mediană a setului

de date şi este utilizat pentru clasificarea datelor în elemente interioare care satisfac modelul (mulţimea de consens/inliers) şi date extreme (outliers).

Se alege pragul de distanţă (residual_treshold), care va determina seturi diferite ale mulţimii de consens (inliers), în funcţie de mărimea pragului aleasă în mod iterativ. Rezultatele diverselor iteraţii sunt vizualizate într-un grafic scatter, care adaptează dreapta de regresie în funcţie de valoarea pragului de distanţă.

3.4. Regresia Decision Tree şi Regresia Random Forest

Random forest este un model de învăţare de ansamblu (ensemble learning). Învaţarea de ansamblu presupune aplicarea mai multor algoritmi de machine learning sau a aceluiaşi algoritm de mai multe ori, pentru a face predicţii mai precise asupra unui set de date. Algoritmul random forest este un model de învaţare supervizată, care poate fi utilizat pentru clasificare sau regresie.

Random forest construieşte în paralel o multitudine de arbori de decizie (decision trees) pe datele de antrenare şi combină rezultatul predicţiilor tuturor arborilor de decizie pentru a elabora rezultatul final [27] [28].

Arborele de decizie, care este un model non-liniar, construieşte modele de clasificare pentru date discrete, sau modele de regresie pentru valori numerice continue, în forma unui arbore.

Un arbore de decizie este un discriminator de categorii, care divide/ramifică în mod recursiv datele de antrenament prin intermediul unor noduri interne, până se obţin nodurile de decizie/frunzele, care reprezintă rezultatele finale:

- ***nodul rădăcină*** este nodul origine de unde se ramifică toate ramurile şi care nu are intrare, doar ieşiri;
- ***nodurile de decizie*** sunt noduri interne, care au o intrare de la nodul rădăcină sau de la alte noduri interne şi au câte două ramificaţii; fiecare nod reprezintă un punct de partiţionare în funcţie de un atribut (testarea după un anumit atribut) care determină modul de divizare al nodului respectiv;
- fiecare ***ramură*** reprezintă rezultatul testului;
- ***frunzele***, care au câte o intrare de la nodul rădăcină sau de la noduri interne şi nu au nicio ramificaţie, sunt noduri finale şi reprezintă clase/categorii obţinute după

calcularea tuturor atributelor; conţinutul nodurilor frunze reprezintă rezultatele modelului.

În cazul în care avem ***o singură variabilă predictor*** x, algoritmul urmează urmatoarele etape:

Intervalul de valori al variabilei predictor x (axa Ox) este împărţit în subseturi mai mici (regiuni sau segmente), deliminate de praguri, notate cu R_i. Pentru fiecare regiune se calculează media valorilor lui y cuprinse în regiunea respectivă, care reprezintă valoarea estimată a variabilei y, respectiv $\hat{y}$.

$$\hat{y} = media\ valorilor\ variabilei\ y$$

Astfel, pentru fiecare segment în parte, variaţia reziduală a erorilor este considerată a fi suma pătratelor diferenţei dintre valorile lui y şi media lui y. În cadrul fiecărui segment se urmăreşte minimizarea sumei pătratelor erorilor SSE.

Se calculează suma pătratelor erorilor pentru fiecare segment/regiune şi se găseşte segmentul care are cea mai mică valoare a SSE. Acest segment devine nodul rădăcină din care se ramifică toate celelalte noduri de decizie (ramificare binară < decât valoarea prag şi respectiv ≥ decât valoarea prag) [29].

$$nodul\ rădăcina = \min\{SSE_i \mid i \in R_i\} = \min\{\sum (y_j - \hat{y}_j)^2 \mid i \in R_i\}$$

Algoritmul se repetă recursiv până la nivelul nodurilor finale (frunzelor), care nu mai pot fi divizate binar în grupuri/categorii mai mici. Fiecare frunză reprezintă media variabilei y, a unui segment (cluster) din totalul observaţiilor.

Algoritmul se va opri atunci când în segmentul R rămâne un singur punct care nu mai poate fi divizat. Se poate stabili iterativ numărul maxim de divizări sau ca nodurile să se dividă până când în fiecare nod sau regiune se regăseşte un număr minim de observaţii, sub care nodurile nu se mai pot diviza. Numărul de niveluri ale nodurilor de decizie şi ale nodurilor frunze reprezintă adâncimea arborelui.

Dacă adâncimea arborelui e mare sau dacă pragul de divizare stabilit este mic (numărul minim de observaţii dintr-o regiune până la care se poate diviza nodul), atunci algoritmul se ajustează mai bine pe datele de antrenare, dar generează overfitting şi o variaţie mare pe datele de test. Reducerea overfittingului se poate face prin stabilirea unei adâncimi mai mici a arborelui sau a unor valori mai mari pentru pragul de divizare.

În cazul în care avem ***mai multe variabile predictor***, paşii de mai sus se vor relua pentru fiecare variabilă. Se va calcula pentru fiecare variabilă predictor segmentul cu cea mai mică valoare a SSE, şi pentru fiecare variabilă atribut acest segment va deveni candidatul pentru a fi nodul rădăcină al arborelui [29].

Se compară candidaţii, iar atributul care vine cu cea mai mică valoare a SSE este atributul câştigător. Segmentul prezentat de atributul câştigător, devine nodul rădăcină al arborelui, care va fi construit în continuare prin compararea celor mai mici valori ale SSE pentru fiecare predictor şi repetând procedura pentru observaţiile rămase, până când acestea nu mai pot fi divizate în grupuri mai mici [29].

Principalele dezavantaje ale modelului arborelui de decizie sunt urmatoarele:

- modelul este sensibil la valorile extreme (outliers)
- se ajustează foarte bine pe datele de antrenament, dar nu performează la fel de bine pe datele de test, având un bias mic pe datele de antrenare, dar o varianţă mare pe datele de test
- modelul rezultat nu poate fi extrapolat pentru valori aflate în afara minimului sau maximului valorilor observate

Pentru evitarea overfittingul-ui pe datele de antrenare se poate utiliza modelul ***random forest***, care construieşte în paralel o multitudine de arbori de decizie pe datele de antrenament, reducând proprietatea de overfitting a algoritmului arborilor de decizie. Random forest construieşte mai mulţi arbori, care sunt instruiţi pe diverse subseturi ale datelor de antrenament, în scopul reducerii varianţei pe datele de test. Cu cât creşte numărul arborilor de decizie construiţi, cu atât scade varianţa. Astfel modelul pierde din interpretabilitatea pe datele de antrenare, dar are o performanţă îmbunătăţită pe datele de test.

Random forest foloseşte tehnica bagging (bootstrap aggregation), care stabileşte în mod aleator, cu înlocuire, subseturi ale setului de date de antrenare, fiecare subset de date fiind utilizat pentru antrenarea unui arbore de decizie asociat [27] [29]. Numărul arborilor de decizie care vor fi construiţi se stabileşte în mod iterativ.

Pentru fiecare observaţie nouă dintr-un set de test, fiecare dintre arborii de decizie va face o predicţie cu privire la valoarea lui y. Valoarea finală a lui y estimată de algoritm va fi reprezentată de media rezultatelor tuturor arborilor individuali [29].

Algoritmul random forest este foarte performant în analiza de regresie şi are multiple avantaje întrucât este foarte stabil, lucrează atât cu variabile numerice cât şi cu varibile categoriale, nu este afectat de variabile nescalate şi nu este afectat de bias.

Principalul dezavantaj al algoritmului este dat de însăşi complexitatea sa, întrucât necesită resurse computaţionale şi resurse de timp pentru antrenare mai mari faţă de alţi algoritmi. Un alt dezavantaj este faptul că modelul rezultat nu poate fi extrapolat pentru valori aflate în afara minimului sau maximului valorilor observate [30].

3.5. Evaluarea performanţei algoritmilor de machine learning

În modelarea datelor sunt utilizate tehnici de învăţare automată supervizată (supervised learning), întrucât setul de date conţine date de intrare, iar rezultatele sunt cunoscute. Astfel sunt implementaţi algoritmi care printr-un proces de instruire pe baza datelor existente, să facă predicţii, care să fie îmbunătăţite ca urmare a analizei performanţei aplicării modelelor respective. Procesul de instruire continuă până când se atinge un nivel de precizie optim. Datele de intrare sunt disponibile sub formă de vectori sau matrici, iar prin aplicarea unor funcţii matematice asupra datelor de antrenare, algoritmii învaţă să facă predicţii pe seturi noi de date, care nu au făcut parte din datele de antrenament.

În modelarea datelor însă nu este suficient să aplicăm diverse tehnici de machine learning, ci trebuie să şi evaluăm performanţa algoritmilor implementaţi, atât pentru a vedea care dintre modele este cel mai adecvat pentru prelucrarea datelor, cât şi pentru a valida rezultatele obţinute, prin compararea eficienţei modelelor aplicate.

În scopul evaluării performanţei modelelor, se pot utiliza tehnici de evaluare a modelelor şi strategii de validare a modelelor.

Pentru atingerea acestor obiective, pot fi utilizate următoarele metode:

- determinarea semnificativităţii modelului în explicarea relaţiei între variabile
- măsurarea erorilor prin diverse tehnici statistice
- implementarea mai multor algoritmi de machine learning şi compararea lor performanţelor lor
- împărţirea datelor în seturi diferite pentru date de antrenare şi date de testare
- validarea încrucişată (k-fold cross-validation)
- compensarea bias-varianţă prin strategii de regularizare a modelelor de regresie liniară

3.6. Metode de evaluare a performanţei modelelor

3.6.1. Coeficientul de determinare

Coeficientul de determinare arată care este proporţia sau procentul din variaţia variabilei y care este explicată de variabila x, arătând care este proporţia în care relaţia de variabilitate dintre x şi y este explicată de modelul de regresie. Coeficientul de determinare este esenţial în determinarea performanţei modelelor liniare.

În cazul regresiei liniare simple, coeficientul de determinare este coeficientul de corelaţie Pearson la pătrat, prin urmare testarea coeficientului de determinare are sens doar dacă coeficientul de corelaţie este semnificativ:

$$R^2 = r_{x,y}^2$$

Pentru ca funcţia de regresie aleasă sa fie semnificativă, trebuie ca variaţia reziduală să fie minimă. Prin urmare, coeficientul de determinare se calculează pe baza următorilor indicatori [19] [17] [18]:

- $\boldsymbol{SSR}$ – variaţia totală a datelor $\hat{y}$ estimate sau suma pătratelor abaterilor de regresie, unde $\hat{y}$ sunt valorile estimate si $\bar{y}$ este media datelor de intrare:

$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

- $\boldsymbol{SSE}$ – variaţia între datele de intrare şi cele estimate sau suma pătratelor valorilor reziduale, unde y sunt datele de intrare, iar $\hat{y}$ sunt valorile estimte prin aplicarea modelului:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- $\boldsymbol{SST}$ – variaţia totală a datelor y de intrare sau suma pătratelor abaterilor individuale de la medie, unde y sunt datele de intrare, iar $\bar{y}$ este media datelor de intrare:

$$\text{SST(variaţia totală)} = \text{SSR(variaţia explicată)} + \text{SSE(variaţia neexplicată)}$$

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

Coeficientul de determinare se calculează după urmatoarea formulă:

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

Coeficientul de determinare ia valori între 0 şi 1 şi se interpretează procentual:

- Dacă $SSE = 0$ şi $R^2 = 1$ înseamnă că modelul regresiei e perfect, variaţiile lui y sunt explicate complet de x şi deci între cele două variabile există o legătură liniară;
- Dacă $SSE = 1$ şi $R^2 = 0$ înseamnă că modelul regresiei nu explică o relaţie între variabile, respectiv între cele două variabile nu există o legătură liniară.

Complementul lui R^2, respectiv $(1 - R^2)$ se numeşte coeficient de nedeterminare şi arată proporţia în care variabila y nu este explicată de variabila x, ci de alţi factori care nu sunt luaţi în considerare de model.

3.6.2. Coeficientul de determinare ajustat

Coeficientul de determinare ajustat este esenţial în evaluarea modelelor de regresie liniară multiplă. Coeficientul de determinare R^2 nu poate arăta dacă predicţiile sunt influenţate de bias. Dacă un model are prea mulţi predictori, este posibil să fie supraestimat (overfitting) şi să îşi piardă abilitatea de a face predicţii pe seturi noi de date.

În cazul regresiei multiple, întrucât adăugând modelului noi variabile predictor, valoarea SSE va scade, implicit valoarea R^2 va creşte. Coeficientul de determinare R^2 creşte cu fiecare variabilă predictor adăugata modelului, chiar dacă unele variabile nu sunt esenţiale în explicarea modelului.

Coeficientul de determinare ajustat R^2_{adj} este corectat cu gradele de libertate, astfel ca adăugând noi variabile neesenţiale în explicarea modelului, acest indicator va scade, aducând o penalizare pentru utilizarea de variabile inutile. Dacă se adaugă modelului variabile utile, atunci valoarea R^2_{adj} va creşte [31].

Coeficientul R^2_{adj} va fi întotdeauna mai mic sau egal cu R^2 şi se calculează după următoarea formulă:

$$R^2_{adj} = 1 - [\frac{(1 - R^2) * (n - 1)}{n - k - 1}]$$

Coeficientul R^2_{adj} permite compararea puterii de predictibilitate a modelelor de regresie care conţin numere diferite de predictori, deoarece arată dacă un număr mai mare de predictori explică mai bine modelul sau nu. Valoarea lui e întotdeauna mai mică decât valoarea lui R^2 [31].

3.6.3. Eroarea medie patratică (mean squared error)

- reprezintă media sumei pătratelor diferenţelor între valorile originale ale variabilei y şi valorile estimate $\hat{y}$ [32] [33] [34]
- măsoară variaţia erorilor
- un scor bun al MSE tinde spre zero, dar evaluarea depinde de specificul setului de date
- este sensibilă la valori extreme (outliers)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

3.6.4. Eroarea rădăcinii medie pătratică (root mean squared error)

- este rădăcina pătrată a erorii medii pătratice [32] [33] [34]
- măsoară abaterea/deviaţia standard a erorilor
- valorile sunt scalate la nivelul variabilei estimate $\hat{y}$, prin urmare vom avea erori cu aceeaşi unitate de măsură ca şi datele originale
- un scor bun al $RMSE$ tinde spre zero, dar evaluarea depinde de specificul setului de date

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

3.6.5. Media erorilor absolute (mean absolute error)

- reprezintă media în valori absolute a diferenţei între valorile actuale a variabilei y şi valorile estimate $\hat{y}$ [32] [33] [34]
- măsoară media valorilor reziduale
- prin comparare cu media variabilei y putem afla cât la sută din medie este MAE; cu cât procentul e mai mic, cu atât modelul e mai precis
- are unităţi de măsură similare cu datele originale, deci poate fi comparată doar între modele ale căror erori sunt măsurate în scală similară
- nu este sensibilă la valori extreme (outliers), deci este mai robustă decat MSE

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

3.6.6. Graficul rezidurilor din regresie

Graficul valorilor reziduale este un instrument foarte util de vizualizare a tiparelor variaţiei rezidurilor şi se construieşte luând pe axa ox variabila x, iar pe axa oy valorile reziduale, care se calculează ca diferenţă între valorile actuale ale variabilei y şi valorile estimate ale acesteia prin aplicarea modelului [19] [20].

Din forma norului de puncte se poate observa distribuţia valorilor reziduale care validează ipoteza modelului regresiei liniare, respectiv:

- dacă există un pattern al acestora înseamnă că există o corelaţie între variabila x şi reziduri, şi relaţia nu poate fi modelată prin regresia liniară
- dacă sunt dispuse la întâmplare fără să formeze un anumit model, înseamnă că variabila x si reziduurile sunt independente, şi relaţia poate fi modelată prin regresia liniară

3.6.7. Testarea modelului de regresie pe baza statisticii t Student

Testul t Student se foloseşte în regresia liniară pentru testarea semnificativităţii statistice a coeficientului de regresie/panta de regresie, în determinarea existenţei unei legături liniare între variabila predictor x şi variabila dependentă y.

În acest sens, se formulează ***ipoteza nulă*** $\boldsymbol{H_0}$, prin care se consideră că variabila y nu este influenţată de variaţia variabilei x şi deci coeficientul b din ecuaţia de regresie nu este semnificativ diferit de zero. ***Ipoteza alternativă*** $\boldsymbol{H_1}$ presupune că variabila y este influenţată de variabila x şi prin urmare valoarea coeficienţilor de regresie este semnificativ diferită de zero [19] [20].

$$\boldsymbol{H_0: b = 0}$$

$$\boldsymbol{H_1: b \neq 0}$$

În cazul regresiei multiple:

$$\boldsymbol{H_0: b_1, b_2, \dots b_m = 0}$$

$$\boldsymbol{H_1: \exists j = \overline{1, i},\ b_j \neq 0}$$

Dacă se respinge ipoteza H_0, cu un prag de semnificaţie α ales, înseamnă că între variabila/variabilele independente şi variabila dependentă există o legătură semnificativă. Pragul de semnificaţie arată probabilitatea de a obţine datele observate pornind de la premisa că ipoteza nulă este adevărată. În practică se consideră de regulă un prag de semnificaţie α=0.05, adică un risc de 5% de a se respinge în mod eronat ipoteza H_0, atunci când este adevărată.

Pragul de semnificaţie de 0.05 în testarea ipotezei nule, a fost stipulat de Fisher, părintele statisticii moderne.

Verificarea ipotezei H_0 se face cu ajutorul testului t Student [19] [18] [20], după formula:

$t = \frac{\hat{b}-b_0}{SE_{\hat{b}}} \sim T_{n-2}$, t este o statistică Student cu (n-2) grade de libertate

unde: $\hat{b}$ este coeficientul de regresie

b_0 este valoarea asumată 0 conform ipotezei H_0

$SE_{\hat{b}}$ este abaterea standard a coeficientului de regresie

iar abaterea standard a coeficientului de regresie este:

$$SE_{\hat{b}} = \frac{\sqrt{\frac{1}{n-2} * \sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

astfel:

$$t = \frac{(\hat{b} - b_0) * \sqrt{n-2}}{\sqrt{\frac{SSE}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}}}} = \frac{(\hat{b} - b_0) * \sqrt{n-2}}{\sqrt{1-\hat{b}^2}}$$

O altă metodă de determinare a lui t este pe baza coeficientului de corelaţie a lui Pearson [20], după formula:

$$t = \frac{r_{x,y} * \sqrt{n-2}}{\sqrt{1-R^2}}$$

Valoarea calculată a lui t se compară cu valoarea teoretică a lui t, obţinută din tabelul Student, pentru (n-2) grade de libertate şi pentru nivelul de semnificaţie α stabilit. Pentru aflarea valorii teoretice se utilizează un risc de α/2, întrucât distribuţia Student este simetrică, iar suprafaţa de respingere α este împărţită în două părţi egale [19] [20].

Dacă valoarea absolută a testului t este mai mare decât valoarea teoretică a testului, se respinge ipoteza nulă, coeficientul de regresie fiind considerat semnificativ pentru explicarea modelului.

3.6.8. Testarea modelului de regresie pe baza statisticii test *F*

Testul *F* Fisher este un test global de semnificaţie a ansamblului coeficienţilor de regresie. Spre deosebire de testul t Student, care se poate aplica doar pentru un coeficient de regresie

odată, testul *F* se poate aplica pentru coeficienţi multipli de regresie simultan, ceea ce permite compararea între modele liniare cu număr diferit de coeficienţi de regresie.

Testul *F* compară modelul care se implementează, cu un model fără variabile independente, deci care are doar intercept, fără a avea coeficienţi de regresie şi verifică dacă coeficienţii de regresie îmbunătăţesc modelul.

Modelul redus bazat doar pe intercept, sugerează că fiecare valoare a variabilei y este o funcţie compusă dintr-o medie generală reprezentată de intercept (media lui y) şi o eroare ε:

Prin urmare ipotezele sunt următoarele [35] [36]:

$\boldsymbol{H_0}$**:** $\boldsymbol{b = 0}$ **(*ipoteza nulă)*** **-** modelul fără variabile independente este la fel de semnificativ ca şi modelul implementat cu variabile independente:

$$y_i = a + \varepsilon_i$$

$\boldsymbol{H_1}$**:** $\boldsymbol{b \neq 0}$ **(*ipoteza alternativă)*** **-** modelul implementat cu variabile independente este mai semnificativ decat modelul bazat doar pe intercept: ***(ipoteza alternativă)***

$$y_i = a + b_1 * x_1 + \cdots b_i * x_i + \varepsilon_i$$

Testul *F* se poate calcula pe baza tabelului ANOVA din Figura 3.1, ca raport între varianţa explicată şi varianţa neexplicată [19] [20] [17]:

Sursa variaţiei	**Suma pătratelor**	**Grade de libertate**	**Media pătratelor**	**F**
Regresie	$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$	$k - 1$	$MSR = \frac{SSR}{k-1}$	$\frac{MSR}{MSE}$
Eroare	$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} e_i^2$	$n - k$	$MSE = \frac{SSE}{n-k}$	
Total	$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$	$n - 1$	$\frac{SST}{n-1}$	

Figura 3.1: Tabelul ANOVA

$$F = \frac{variatia\ explicata}{variatia\ neexplicata} = \frac{SSR}{SSE} * \frac{n-k}{k-1} \sim F_{k-1,n-k}$$

Valoarea *F* obţinută se compară cu valoarea teoretică (valoarea critică) din tabelul F, care se citeşte în funcţie de: nivelul de semnificaţie α stabilit şi de gradele de libertate (k-1, n-k). Rezultatul este semnificativ dacă valoarea testului F este mai mare decât valoarea teoretică citită din tabelul F, caz în care se respinge ipoteza nulă şi se acceptă ipoteza alternativă, considerându-

se că modelul cu coeficient de regresie este mai semnificativ decât modelul bazat doar pe intercept.

3.7. Strategii de validare a modelelor

3.7.1. Validarea simplă (seturi de date disjuncte de antrenare şi testare)

În scopul eliminării supraestimării (overfitting), datele pot fi împărţite în date de antrenare şi date de testare. Nu este eficient să antrenăm algoritmii pe un set de date şi să utilizăm exact acelaşi set de date pentru a evalua algoritmii. Dacă utilizăm un algoritm pe un set de date pentru antrenare şi utilizăm acelaşi set pentru testare, atunci algoritmul va avea un scor perfect pe setul de date de antrenament, dar nu ştim cum se va comporta modelul aplicat altor seturi de date, care nu apar în setul de antrenament. Este foarte posibil ca performanţa aplicării aceluiaşi model pe alte seturi de date să fie foarte slabă. Din acest motiv, este indicat să se facă testări ale modelului implementat pe un set de date de testare, diferit de setul utilizat pentru datele de antrenare [37].

Prin compararea rezultatelor şi a scorului obţinut prin aplicarea algoritmului asupra datelor de antrenament, cu cele obţinute prin aplicarea modelului asupra datelor de testare, se poate estima dacă modelul nu este afectat de overfitting, dacă nu este afectat de variabile neluate în calcul (de ex. date extreme – outliers – care pot altera rezultatele) şi deci dacă modelul este eficient pentru utilizare operaţională.

Pe baza datelor de antrenare se estimează parametrii funcţiilor de regresie, iar pe baza datelor de testare se evaluează modelele implementate.

3.7.2. Validarea încrucişată (k-fold cross validation)

În scopul evaluării performanţei modelelor implementate se poate folosi validarea încrucişată, care efectuează mai multe iteraţii pe acelaşi set de date, prin împărţirea setului în sub-seturi egale de date, după cum urmează [37] [38]:

- setul de date este împărţit în k părţi sau grupuri egale (de aici denumirea de k-fold), k-1 subseturi pentru antrenare şi 1 set pentru testare (dimensiunea unui sub-set = dimensiunea setului / k)
- algoritmul este antrenat pe k-1 părţi şi testat pe o parte din cele k stabilite
- se repetă antrenarea şi testarea de k ori, pentru toate variantele de împărţire a grupurilor

Alegerea numărului k se face în funcţie de mărimea setului de date, astfel că dimensiunea fiecărui grup sau partiţie să fie suficient de mare pentru a fi reprezentativă.
Un exemplu de validare încrucişată pentru k = 5 este prezentat în Figura 3.2:

Împărţirea 1	Grupul 1 testare	Grupul 2 antrenare	Grupul 3 antrenare	Grupul 4 antrenare	Grupul 5 antrenare	=>	Scorul 1
Împărţirea 2	Grupul 1 antrenare	Grupul 2 testare	Grupul 3 antrenare	Grupul 4 antrenare	Grupul 5 antrenare	=>	Scorul 2
Împărţirea 3	Grupul 1 antrenare	Grupul 2 antrenare	Grupul 3 testare	Grupul 4 antrenare	Grupul 5 antrenare	=>	Scorul 3
Împărţirea 4	Grupul 1 antrenare	Grupul 2 antrenare	Grupul 3 antrenare	Grupul 4 testare	Grupul 5 antrenare	=>	Scorul 4
Împărţirea 5	Grupul 1 antrenare	Grupul 2 antrenare	Grupul 3 antrenare	Grupul 4 antrenare	Grupul 5 testare	=>	Scorul 5

Figura 3.2: Tabel validare încrucişată

Ca urmare a executării tuturor iteraţiilor, fiecare grup de date va fi utilizat de k-1 ori pentru antrenare şi o dată pentru testare.

Validarea încrucişata aplicată diverşilor algoritmi de machine learning utilizaţi, permite compararea modelelor pentru a se determina eficienţa aplicării acestora asupra setului de date, respectiv performanţa algoritmilor.

3.7.3. Compensare bias - varianţă

În evaluarea eficienţei unui model de machine learning este importantă obţinerea unui compromis între bias-ul şi variaţia parametrilor estimaţi, compromis care contribuie la implementarea de modele adecvate pentru datele analizate, precum şi la evitarea greşelilor de overfitting (supraestimare) şi underfitting (subestimare). Bias-ul şi variaţia parametrilor estimaţi sunt importante surse de erori, care trebuie minimizate pentru a putea implementa algoritmi de învaţare automată supervizată care să poată fi aplicaţi unor seturi diferite de date şi deci care să fie valabili dincolo de datele de antrenare [39] [40].

Bias-ul înseamnă incapacitatea unui model de machine learning de a descrie şi reprezenta în mod fidel relaţia între date. Un model cu un bias mare este un model care întrucăt nu se poate ajusta pe datele de antrenament, suprasimplifică descrierea modelului, ceea ce determină o valoare mare a variaţiei erorilor, astfel că principiul minimizării sumei pătratelor erorilor nu poate fi respectat [39] [29] [41].

De exemplu dacă aplicăm un model liniar asupra unor date distribuite sub formă de curbă, vom avea underfitting, întrucât modelul liniar va simplifica reprezentarea printr-o dreaptă şi va

genera o valoare mare a sumei pătratelor erorilor. Pentru a evita underfittingul vom creşte complexitatea modelului. Astfel, un model polinomial de grad mai mare ar avea probabil un bias minim şi o variaţie a erorilor apropiată de zero, reprezentând foarte fidel relaţia între date **($SSE \to 0$ si $R^2 \to 1$**, ceea ce arată un model perfect pentru explicarea datelor, în care suma pătratelor erorilor este 0).

Variaţia înseamnă diferenţa între rezultatele obţinute prin aplicarea modelului pe datele de antrenare şi cele obţinute din aplicarea modelului pe datele de testare. Modelele cu variaţie mare reprezintă foarte bine datele de antrenament, cu media pătratelor erorilor aproape de zero, dar înregistrează erori mari la aplicarea pe datele de testare [39] [41] [29].

De exemplu un model polinomial de grad mai ridicat aplicat datelor distribuite sub formă de curbă, ar avea un bias foarte aproape de zero pe datele de antrenare, dar datorită overfittingului, ar genera o variaţie a erorilor foarte mare, mult mai mare decât modelul liniar, atunci când ar fi aplicat datelor de testare. **($SSE \to 1$ si $R^2 \to 0$**, faţă de valorile aplicate setului de antrenare, care arată un model perfect).

În concluzie:

- dacă modelul implementat este prea simplu şi are prea puţini parametrii, va avea un bias mare la datele de antrenare, dar va avea o variaţie mai mică decât modelul complex la datele de testare
- dacă modelul implementat este prea complex şi are un număr prea mare de parametrii va avea un bias mic la datele de antrenare, dar va avea o variaţie mare la datele de testare

Prin urmare, pentru a avea un algoritm eficient care să modeleze în mod fidel relaţia între date şi care să producă predicţii consistente prin aplicarea asupra unor seturi diferite de date, este nevoie să existe un compromis între bias-ul şi variaţia parametrilor estimaţi.

Compromisul dintre bias-ul şi variaţia parametrilor estimaţi constituie baza conceptuală a metodelor de regularizare (optimizare) a regresiei, cum sunt ***regresia lasso*** şi ***regresia ridge***. Metodele de regularizare introduc în soluţiile de regresie conceptul de bias, care poate reduce variaţia parametrilor în mod considerabil, faţă de metoda celor mai mici pătrate.

3.7.4. Regresia Ridge

Regresia ridge (regularizare L2) presupune calcularea unei noi linii de regresie, care nu se potrivește perfect datelor de antrenare, prin introducerea unei cantităţi mici de Bias, numită factor de regularizare, cu scopul de a evita supraînvăţarea modelului (overfitting). Chiar dacă

linia de regresie ridge nu reprezintă perfect estimatorii ca în cazul metodei celor mai mici pătrate, regresia ridge asigură predicţii mai generalizate şi pentru seturi de date noi, nu doar pentru cele de antrenare, întrucât determină scăderea variaţiei erorilor la un nivel semnificativ.

Regresia ridge adaugă o penalizare funcţiei de cost (media pătratelor erorilor) a modelului regresiei liniare bazat pe minimizarea pătratelor erorilor [41] [29]:

$$mse = \frac{1}{n}\sum(y_i - \hat{y}_i)^2 + \lambda * \sum b^2$$

unde: b^2 este penalitatea L2 la metoda celor mai mici pătrate, reprezentată de pătratul coeficientului de regresie

λ determină mărimea penalităţii şi poate lua valori cuprinse între 0 şi infinit

Atunci când $\lambda = 0$ => dreapta de regresie liniară este egală cu dreapta de regresie ridge. La fiecare creştere a lui λ, panta devine tot mai puţin abruptă şi se apropie asimptotic de axa ox, astfel că variabila dependentă y devine tot mai puţin sensibilă la modificările variabilei independente x.

Regresia ridge îmbunătăţeşte predicţiile pentru seturi noi de date întrucât reduce variaţia erorilor pentru datele de test, făcând predicţia mai puţin sensibilă faţă de datele de antrenare.

În cazul regresiei liniare multiple toţi parametrii sunt regularizaţi în mod egal, factorul de regularizare fiind produsul dintre λ şi suma pătratelor parametrilor.

Valoarea lui λ se determină prin tehnica validării încrucişate. Astfel, se încearcă mai multe valori pentru λ şi se foloseşte validarea încrucişată pentru a determina care variantă are cea mai mică variaţie a erorilor şi cel mai mare scor al coeficientului de determinare R^2.

3.7.5. Regresia Lasso

Regresia lasso (regularizarea L1) este similară cu regresia ridge, prin introducerea unui factor de regularizare. În acest caz, penalitatea L1 este reprezentată de valoarea absolută a coeficientului de regresie [41] [29]:

$$mse = \frac{1}{n}\sum(y_i - \hat{y}_i)^2 + \lambda * \sum |b|$$

Diferenţa între regularizarea L2 şi L1 este aceea că, în cazul regresiei ridge panta se apropie asimptotic de 0, fără să ajungă la valoarea 0, iar în cazul regresiei lasso panta se apropie de 0, până la valoarea 0.

În cazul regresiei liniare multiple, regresia lasso poate duce la excluderea anumitor variabile care nu sunt necesare ecuaţiei de regresie, ducând la reducerea variaţiei în modelele care conţin variabile inutile în explicarea modelului şi la o ecuaţie finală mai simplă şi mai uşor de interpretat. Astfel, regresia lasso contribuie la minimizarea numărului variabilelor independente în explicarea unui model de regresie liniară multiplă.

3.7.6. Implementarea comparativă a algoritmilor de machine learning

În scopul validării unui model de regresie se pot implementa mai mulţi algoritmi de regresie, asupra cărora au fost aplicate validarea simplă, prin împărţirea setului de date în date de antrenare şi date de testare şi/sau validarea încrucişată, în scopul autoevaluării performanţelor fiecărui algoritm în parte.

Scorurile obţinute prin aplicarea diverselor modele sunt comparate, atât pentru validarea reprezentativităţii fiecărui algoritm în parte, cât şi pentru a identifica care este modelul cel mai eficient în explicarea datelor şi a relaţiilor dintre ele.

4. Studiu de caz – predictia preţului autoturismelor prin analiza de regresie

4.1. Tehnologii utilizate pentru analiza şi modelarea datelor

Pentru prezentul proiect de data science, a fost utilizat Python, împreună cu Jupyter Notebook şi pachetele Numpy, Pandas, Matplotib şi Scikit-learn.

Python

Python este un limbaj de programare dinamic, multifuncţional, de nivel înalt, care face parte din categoria limbajelor de programare interpretate şi care permite atât programarea imperativă, funcţională sau procedurală, cât şi programarea orientată obiect [42] [43].

Limbajele de programare interpretate funcţionează pe baza unui interpretor, adică un program care va interpreta codul scris şi îl va transforma în cod maşină. Codul scris de utilizator va rămâne în mod text, adică nu va fi compilat până la nivel de cod maşină, până la momentul rulării. Din punct de vedere al funcţionalităţii, avantajul unui limbaj interpretat este faptul ca poate rula acelaşi program pe toate platformele unde există un interpretor (dacă scriem un program Python pe Windows, putem să-l rulăm şi pe Linux sau Mac şi invers).

Un mare avantaj al utilizării Python decurge din faptul că are o gamă largă de biblioteci standard de metode sau funcţii implementate care pot fi folosite (lucrul cu fişiere externe, conectare la baze de date, module pentru implementarea de interfeţe grafice, etc.) şi permite extinderea funcţionalităţii prin numeroase pachete adiţionale programate şi puse la dispoziţie de terţi, care sunt create pentru a îndeplini diverse funcţii. În noiembrie 2019, PyPI - Python Package Index (Indexul pachetelor Python) [44], colecţia oficială a pachetelor software create de terţi, conţinea peste 200.000 de pachete, cu o arie largă de funcţionalităţi, printre care: baze de date, analiză de date, interfeţe grafice, procesare de imagini, informatică ştiinţifică, web scraping, machine learning, etc.

Prin toate aceste facilităţi pe care le pune la dispoziţie, Python este un sistem multifuncţional foarte util în data science, deoarece facilitează: colectarea datelor din diverse surse (fişiere externe, baze de date, web scraping), pregătirea datelor (analizarea datelor brute, curăţarea şi prelucrarea lor prin transformarea din date brute în date viabile pentru prelucrare), analiza statistică descriptivă şi inferenţială a datelor, reprezentarea grafică a datelor pentru vizualizarea distribuirii datelor şi a diverselor corelaţii între ele, modelarea datelor prin tehnici de statistică şi machine learning

Jupyter Notebook

Proiectul Jupyter (dezvoltat din IPython începând cu anul 2014) a fost creat pentru dezvoltarea de software open-source şi servicii pentru computing interactiv, şi suportă medii de execuţie pentru mai mult de 40 de limbaje de programare, printre care şi Python.

Proiectul Jupyter a dezvoltat produsul pentru computing interactiv Jupyter Notebook (cunoscut iniţial ca IPython Notebooks), care este o aplicaţie web open-source care permite crearea şi partajarea de documente care conţin cod Python [45]. Un document scris în Jupyter Notebook este un document JSON, care conţine o listă ordonată de celule de input şi output, care conţin cod, text, formule matematice, grafice, etc. Jupyter Notebook permite vizualizare grafică de date, ecuaţii, analiza datelor şi modelare statistică, machine learning, etc. şi este foarte adecvat pentru proiecte de data science datorită interfeţei grafice implicite şi a modului de afişare a datelor, sub formă de jurnal, ca o succesiune de celule [45] [46].

NumPy

NumPy este o bibliotecă pentru Python, care permite lucrul eficient cu serii foarte mari de date prezentate sub formă de vectori şi matrici unidimensionale sau multidimensionale. Pachetul NumPy pune la dispoziţie o colecţie mare de operatori şi funcţii matematice pentru manipularea rapidă a datelor de tip arrays. NumPy permite şi procesarea imaginilor.

Pachetul NumPy, care suportă şi aborarea orientată obiect, permite efectuarea rapidă de operaţii cu matrici, inclusiv manipulare matematică, logică, manipularea formei matricilor prin modificarea liniilor şi coloanelor, sortarea şi selectarea datelor după anumite criterii, operaţii de algebră liniară, operaţii statistice simple, etc [47].

Întrucat seriile de date care se analizează în proiectele de data science se prezintă sub formă de tabele şi sunt procesate sub formă de matrici, pachetul NumPy oferă suportul pentru manipularea eficientă a acestor date, prin faptul că permite transformarea matricilor, selectarea anumitor coloane şi linii, compunerea a două sau mai multe matrici, eliminarea anumitor elemente ale matricilor, etc. astfel ca datele brute analizate să fie transformate după cum este necesar [48].

Pandas

Este o bibliotecă Python, un instrument pentru manipularea şi analiza seriilor de date. Pachetul pune la dispoziţia utilizatorilor structuri de date şi operaţii pentru manipulare de tabele şi analiza time-series, care presupune observarea evoluţiei unor date furnizate de acelaşi eşantion pe diverse perioade de timp [49].

Datele sunt disponibile sub formă de tabele, desfăşurate sub formă de linii şi coloane, iar Pandas le stochează într-un DataFrame, o structură de date 2-dimensională care poate stoca date numerice, caractere, date categoriale, etc. Pachetul conţine instrumente pentru citirea şi scrierea datelor, analizarea datelor lipsă sau a anomaliilor, modificarea tabelelor, obţinerea de sub-seturi de date din seturi mari de date, inserarea sau ştergerea de coloane şi linii în tabele, compunerea unui tabel din mai multe tabele, selectare şi sortare după diverse criterii, filtrare de date, împărţirea datelor în grupuri de date, etc. Totodată, Pandas permite diverse analize statistice pentru date numerice, facilitând calcularea diverşilor indicatori statistici, necesari în analiza datelor [50].

Matplotlib

Matplotlib este o bibliotecă open-source pentru Python pentru vizualizare grafică a datelor cantitative şi statistică descriptivă. Pachetul furnizează o interfaţă API orientată obiect, pentru prezentarea de grafice, în scopul vizualizării statice, animate sau interactive a datelor în Python. Datele pot fi prezentate sub formă de histograme (tabel de frecvenţe), diagrame cu bare, diagrama de procente (pie-chart sau diagrama tip sector circular), diagrame de tip arie sau de tip suprafaţă, boxplots, scatterplot (grafic de dispersie), grafice time-series, etc [51].

Deoarece sunt analizate cantităţi mari de date (în general numerice), în scopul prelucrării, analizării şi comunicării acestor date este necesară reprezentarea grafică a datelor pentru vizualizarea lor, în scopul observării modului de distribuire sau de împrăştiere a datelor, a identificării unor eventuale relaţii de cauzalitate sau de corelaţie între date, pentru a putea identifica diverse modele sau relaţii între date pe baza variabilelor analizate şi pentru a descrie valori codificate ca obiecte vizuale [52].

Scikit-learn (Sklearn)

Scikit-learn este o bibliotecă Python pentru machine learning, concepută pentru a funcţiona în relaţii de interoperabilitate cu bibliotecile NumPy şi SciPy [53]. Scikit-learn utilizează pachetul NumPy pentru operaţii de algebră liniară şi operaţii cu matrici. De asemenea, Scikit-learn se integrează foarte bine cu alte librării Python, printre care Pandas şi Matplotlib.

Scikit-learn facilitează preprocesarea datelor sub formă de array, întrucât furnizează funcţii şi clase pentru transformarea datelor în reprezentări adecvate pentru a fi analizate, tehnici de factorizare a matricilor (analiza în componente principale, analiza factorială, etc). De

asemenea, pe baza calculelor statistice, algoritmii de machine learning pot identifica anomaliile şi extremele dintr-un set de date [54].

Pachetul implementează numeroşi algoritmi pentru învăţare automată supervizată, semi-supervizată şi nesupervizată, respectiv algoritmi de regresie, analiza discriminantă liniară şi quadratică, support vector machines, nearest neighbors, naive Bayes, arbori de decizie, reţele neuronale, clustering (grupare) şi mulţi alţi algoritmi, fiind un suport important pentru proiectele de data science [54].

4.2. Analiza şi modelarea datelor

4.2.1. Analiza preliminară

Setul de date care stă la baza prezentului studiu conţine 205 observaţii şi are 26 de atribute/variabile, de tip numeric şi categorial. După eliminarea unor atribute care nu sunt necesare pentru analiza datelor, respectiv id, simbol, denumire auto, au rămas 22 de atribute, dintre care 14 de tip numeric şi 8 de tip categorial, 21 fiind atributele explicative, conform Figura 4.1:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 22 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   fueltype          205 non-null    object
 1   aspiration        205 non-null    object
 2   doornumber        205 non-null    object
 3   carbody           205 non-null    object
 4   drivewheel        205 non-null    object
 5   wheelbase         205 non-null    float64
 6   carlength         205 non-null    float64
 7   carwidth          205 non-null    float64
 8   carheight         205 non-null    float64
 9   curbweight        205 non-null    int64
 10  enginetype        205 non-null    object
 11  cylindernumber    205 non-null    object
 12  enginesize        205 non-null    int64
 13  fuelsystem        205 non-null    object
 14  boreratio         205 non-null    float64
 15  stroke            205 non-null    float64
 16  compressionratio  205 non-null    float64
 17  horsepower        205 non-null    int64
 18  peakrpm           205 non-null    int64
 19  citympg           205 non-null    int64
 20  highwaympg        205 non-null    int64
 21  price             205 non-null    float64
dtypes: float64(8), int64(6), object(8)
memory usage: 35.4+ KB
```

Figura 4.1: Variabile iniţiale car-price

În scopul efectuării analizei datelor, observaţiile de tip categorial au fost convertite în echivalente numerice prin operaţii de codare. Pentru fiecare atribut au fost calculaţi anumiţi indicatori, respectiv media, abaterea standard, valorile minime şi maxime, şi nu au fost identificate date extreme sau erori de măsurare.

Pentru verificarea şi validarea rezultatelor studiului, datele au fost împărţite în date de antrenare/train data şi date de test/test data. Astfel, din cele 205 observaţii, 70% au fost selectate în mod aleator pentru antrenare şi 30% pentru testare.

Variabilele independente sunt cele 21 de variabile numerotate de la 0-20, iar variabila price este variabila dependentă.

4.2.2. Reducerea dimensionalităţii datelor

Pentru identificarea principalelor atribute/variabile independente care pot explica preţul, au fost utilizate mai multe metode.

Filtrul corelaţiei slabe între predictori şi variabila target

Pentru a explica o variabilă pe baza altor variabile este necesar ca între variabilele independente şi variabila dependentă să existe un anumit nivel de corelaţie. A fost utilizat coeficientul de corelaţie Pearson pentru determinarea variabilelor care au un coeficient de corelaţie cu variabila preţ mai mare decât 0,5, iar variabilele cu un nivel al coeficientului de corelaţie sub 0,5 au fost eliminate.

Numărul variabilelor esenţiale pentru explicarea preţului s-a redus de la 21 la 11, conform Figura 4.2:

```
drivewheel        0.503460
wheelbase         0.550372
carlength         0.673316
carwidth          0.771034
curbweight        0.838979
cylindernumber    0.719352
enginesize        0.876751
boreratio         0.543367
horsepower        0.811184
citympg           0.736946
highwaympg        0.749342
price             1.000000
Name: price, dtype: float64
```

Figura 4.2: Variabile esenţiale car-price

Random Forest pentru determinarea importanţei atributelor

În urma aplicării Random Forest feature importances, cu un scor semnificativ de 0,9850 s-au obţinut următoarele rezultate, conform Figura 4.3:

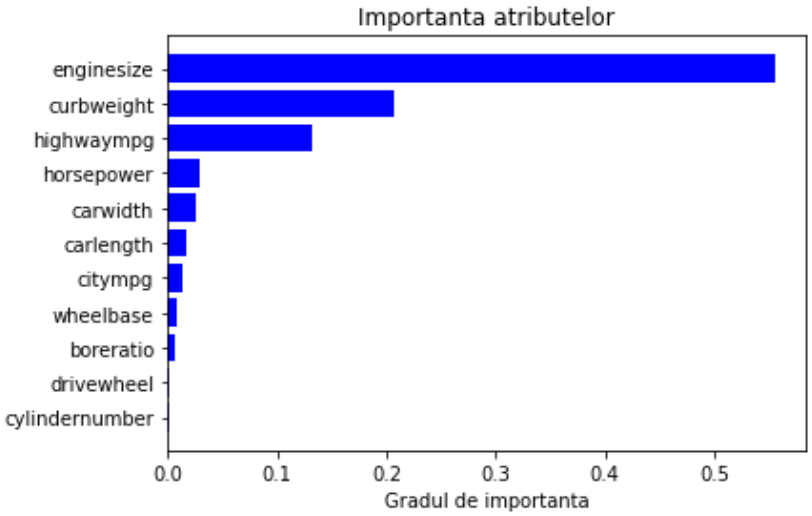

Figura 4.3: Random Forest – importanţa atributelor

Scorul celor mai importante atribute se prezintă astfel:

```
1 0.5550654500067229
2 0.20750191106669158
3 0.1324827167558145
4 0.02972178482112183
```

Este de reţinut faptul că Random Forest va determina cele mai semnificative atribute, dar va subestima atributele care sunt foarte corelate între ele. Prin urmare a fost utilizată încă o metodă de selecţie a atributelor şi au fost comparate rezultatele.

Regresia Lasso pentru selectarea atributelor

Prin aplicarea regresiei lasso cu validare încrucişată a fost determinat cel mai bun coeficient de penalizare având valoarea de 3409,49, iar scorul (coeficientul de determinare) obţinut de model la valoarea de 0,8064 este semnificativ. Rezultalele se prezintă conform Figura 4.4:

Scorul celor mai importante atribute se prezintă astfel:

```
(4.8572, 'curbweight')
(47.5523, 'horsepower')
(75.2475, 'enginesize')
```

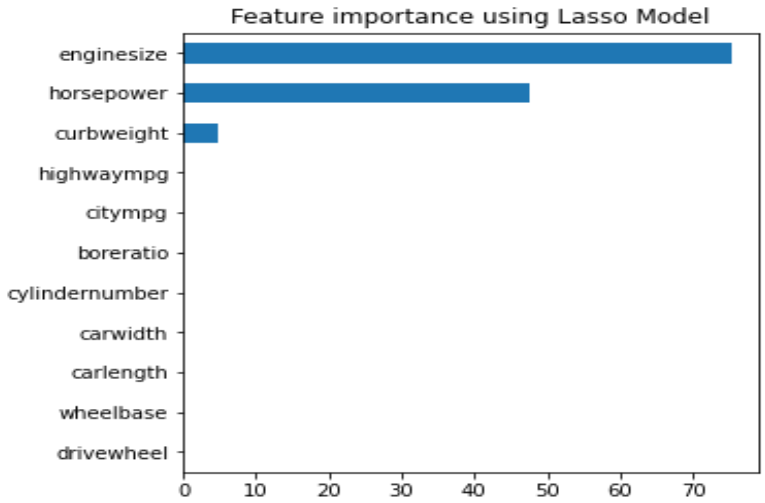

Figura 4.4: Lasso – importanţa atributelor

După cum se poate observa, cele două modele de selecţie a atributelor au rezultate asemănătoare, astfel că au fost reţinute în explicarea modelului variabilele **enginesize, curbweight, horsepower şi highwaympg.**

- ***enginesize*** reprezintă capacitatea cilindrică a motorului şi determină puterea motorului şi consumul de combustibil; se poate exprima în litri, centimetri cubi sau inci cubi;
- ***curbweight*** reprezintă greutatea totală a autovehiculului cu tot echipamentul standard;
- ***horsepower*** sunt caii putere care arată randamentul unui motor;
- ***highwaympg*** reprezintă un indicator al consumului de combustibil pe autostradă; nivelul consumului este determinat de tehnologii complexe şi este dezirabil un consum cât mai mic, atât pentru economisirea resurselor cât şi pentru scăderea emisiilor de CO2.

Prin utilizarea metodelor de reducere a dimensionalităţii datelor, setul de observaţii a fost redus substanţial, prin eliminarea atributelor explicative care nu sunt esenţiale pentru explicarea variabilei target. Astfel, din 25 de atribute explicative aflate în setul iniţial de date au rămas 4 atribute. De la 5.125 observaţii aflate în setul iniţial de date, au rămas pentru modelare 820 de observaţii. Reducerea numărului de observaţii odată cu parcurgerea etapelor de reducere a dimensionalităţii este prezentată în Figura 4.5:

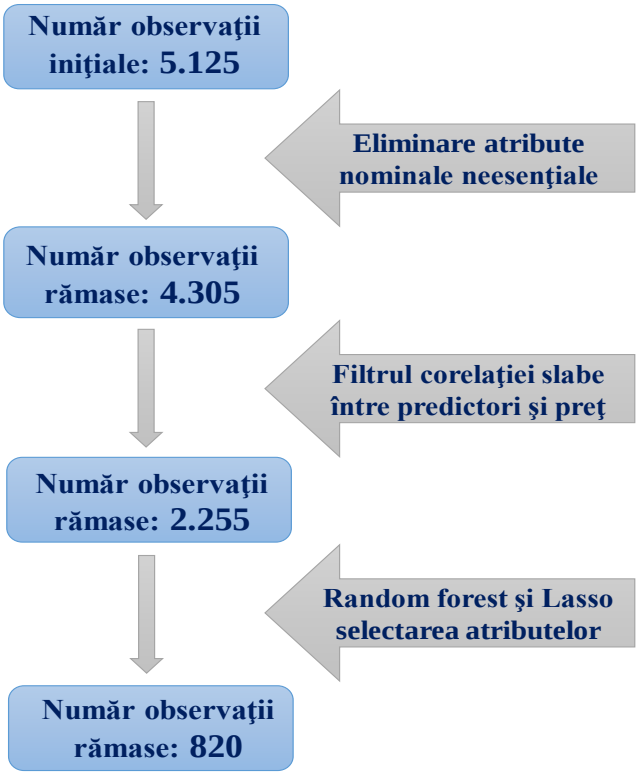

Figura 4.5: Reducerea nr. de observaţii în urma reducerii dimensionalităţii datelor

4.2.3. Determinarea strategiei şi a tehnicilor de modelare

Pentru stabilirea gradului de corelaţie între toate variabilele implicate în model, a fost construită matricea de corelaţie, care poate fi vizualizată în Figura 4.6:

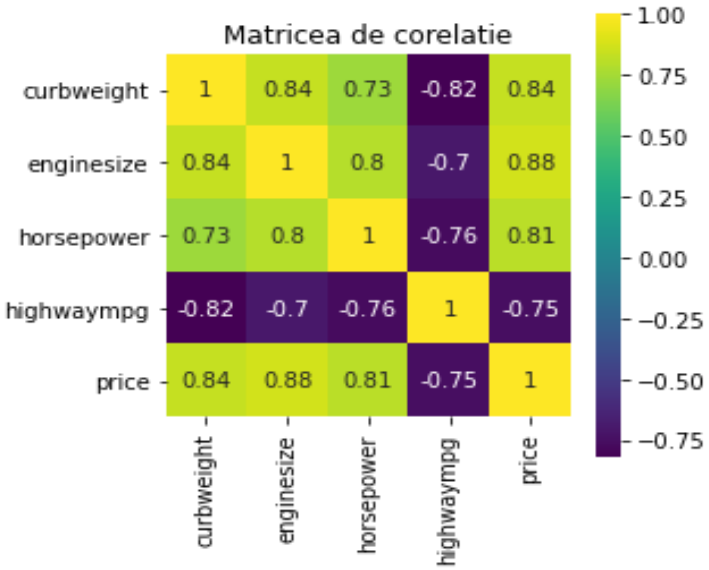

Figura 4.6: Matricea de corelaţie

Se poate observa faptul că între toate variabilele există un nivel de corelaţie ridicat, că toate variabilele au corelaţie pozitivă, în afară de variabila highwaympg, care este corelată negativ cu celelalte variabile, inclusiv cu variabila price. Pentru vizualizarea relaţiilor dintre variabilele implicate în model s-a generat graficul perechilor de variabile, conform Figura 4.7.

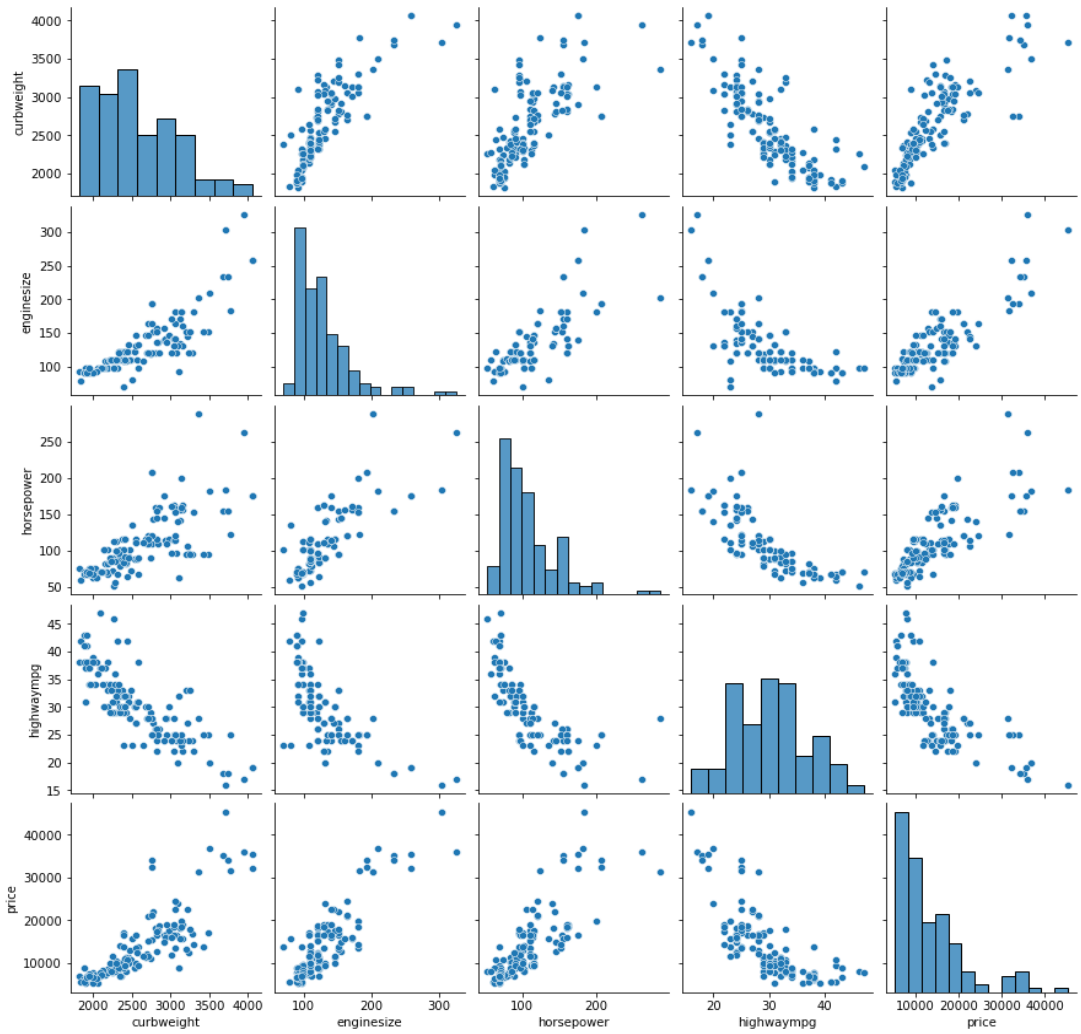

Figura 4.7: Graficul perechilor de variabile

Vizual se observă existenţa unei relaţii care poate fi modelată liniar între variabila independentă enginesize şi preţ, iar între celelalte variabile independente şi variabila target se observă existenţa unor relaţii neliniare.

Având în vedere gradul ridicat de corelaţie între fiecare dintre variabilele independente şi variabila target, şi pentru a respecta principiul multicoliniarităţii, se păstrează variabila care este cea mai corelată cu variabila dependentă, respectiv enginesize şi sunt eliminate celelalte variabile din modelul explicativ. Întrucât se poate observa vizual o relaţie liniară între

enginesize şi preţ, s-a ales modelul regresiei liniare simple, care va fi completat cu regresia lasso şi ridge, dar se va utiliza comparativ şi regresia random forest.

Având în vedere specificul şi scopul prezentului studiu, respectiv acela de a facilita întocmirea unui business plan pentru investiţii, considerăm că varianta teoretică prin utilizarea doar a regresiei simple nu este suficientă pentru explicarea modelului. În demersurile de întocmire a unui plan de business, o companie are nevoie de cât mai multe informaţii şi nu de un model simplificat care este insuficient pentru stabilirea strategiei de producţie.

Astfel, se porneşte de la ipoteza ca un model cu mai multe variabile independente va explica mai bine variabila dependentă şi va aduce mai multe informaţii utile. Datorită acestor considerente se va efectua şi analiza de regresie multiplă, păstrându-se toate cele 4 variabile independente în modelul explicativ.

În concluzie, vor fi implementate modele de regresie liniară, regresia ridge şi lasso, regresia random forest, după cum urmează:

- regresia simplă: x = enginesize → y = preţ;
- regresia multiplă: x_i= enginesize, curbweight, horsepower, highwaympg → y = preţ;
- un model alternativ bazat pe analiza de regresie random forest.

În scopul evaluării performanţelor şi a identificării celui mai semnificativ model, se va face o analiză comparativă a tuturor tehnicilor de modelare eficiente în explicarea preţului.

4.2.4. Regresia liniară simplă

În urma implementării modelului regresiei liniare simple s-a obţinut **ecuaţia funcţiei liniare de regresie**:

$$y = -7646.435077807868 + [165.76866245]\, x$$

Rezultatele modelării prin regresia liniară simpla sunt prezentate în Figura 4.8:

OLS Regression Results

Dep. Variable:	price	R-squared:	0.769
Model:	OLS	Adj. R-squared:	0.767
Method:	Least Squares	F-statistic:	468.6
Date:	Thu, 27 May 2021	Prob (F-statistic):	1.14e-46
Time:	16:57:50	Log-Likelihood:	-1379.9
No. Observations:	143	AIC:	2764.
Df Residuals:	141	BIC:	2770.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-7646.4351	1022.771	-7.476	0.000	-9668.383	-5624.487
enginesize	165.7687	7.658	21.647	0.000	150.629	180.908

Omnibus:	9.817	Durbin-Watson:	2.111
Prob(Omnibus):	0.007	Jarque-Bera (JB):	9.998
Skew:	0.555	Prob(JB):	0.00674
Kurtosis:	3.668	Cond. No.	432.

Figura 4.8: Rezultate regresia liniară simplă OLS

Ipotezele asupra proprietăţilor estimtorilor

Liniaritatea modelului: se poate observa vizual din Figura 4.9, unde se poate observa existenţa unei relaţii liniare între cele două variabile. Coeficientul de corelaţie Pearson cu valoarea de 0,88 confirmă ipoteza existenţei unei relaţii liniare.

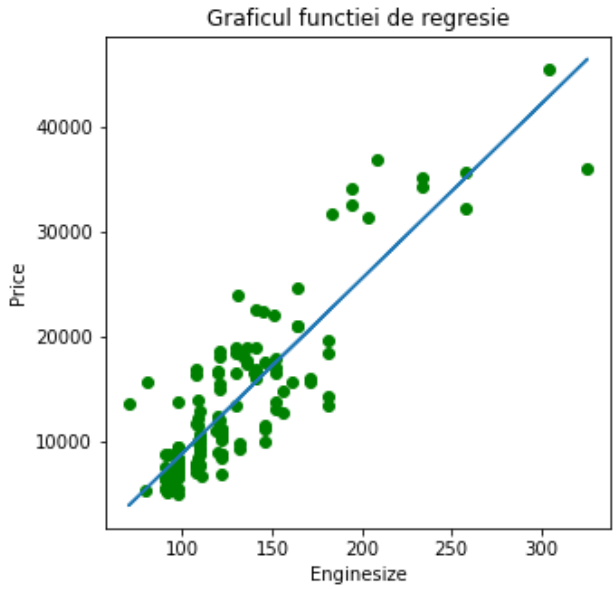

Figura 4.9: Graficul funcţiei de regresie liniară simplă

Media erorilor este 8.1409, fiind foarte apropiată de 0, ceea ce confirmă ipoteza.

Normalitatea erorilor se poate confirma vizual prin histograma erorilor prezentată în Figura 4.10 şi prin verificarea coeficientului de asimetrie (Skew) de 0,555 - apropiat de 0 şi a coeficientului de boltire (Kurtosis) de 3,668 - apropiat de 3. Putem considera că erorile reziduale sunt distribuite normal.

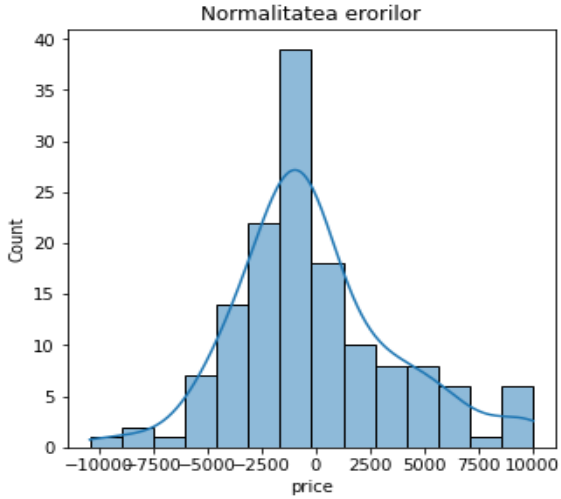

Figura 4.10: Normalitatea erorilor - regresia liniară simplă

Homoscendasticitatea s-a verificat prin reprezentarea grafică a relaţiei dintre variabila x şi reziduuri, prezentată în Figura 4.11:

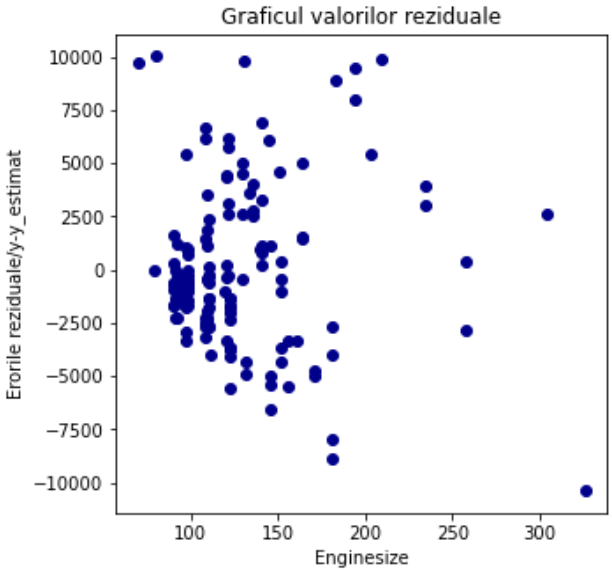

Figura 4.11: Graficul valorilor reziduale – regresia liniară simplă

Se poate observa faptul că punctele nu formează un anumit model sau tipar, astfel ca ipoteza homoscendasticităţii se confirmă.

Necorelarea erorilor a fost verificată prin testul Durbin Watson care are o valoare de 2,111. Pentru α=0,5 limita inferioară citită din tabelul Durbin Watson este de 1,694, iar limita

superioară este 2,346. Întrucât valoarea testului se află în intervalul valorilor teoretice, putem considera ca nu există autocorelare între valorile reziduale.

Testarea semnificativităţii pantei de regresie şi a semnificativităţii globale

Testul t Student: $t_{calculat} = 21{,}647 > t_{teoretic} = 1.14$, se respinge ipoteza nulă.

Testul F: $F_{calculat} = 468{,}6 > F_{teoretic} = 1.14$, se respinge ipoteza nulă.

Coeficientul de determinare de 0,769 obţinut pentru datele de antrenare este semnificativ. De asemenea pe datele de test s-a obţinut un scor de 0,75.

În concluzie, întrucât modelul implementat a trecut testele de evaluare, se poate considera reprezentativ şi poate fi utilizat pentru estimarea variabilei dependente price, în funcţie de variabila independentă enginesize.

A fost dezvoltată o metodă de calcul a ***intervalului de predicţie*** şi a ***intervalului de încredere*** a pantei modelului de regresie, care sunt reprezentate grafic în Figura 4.12:

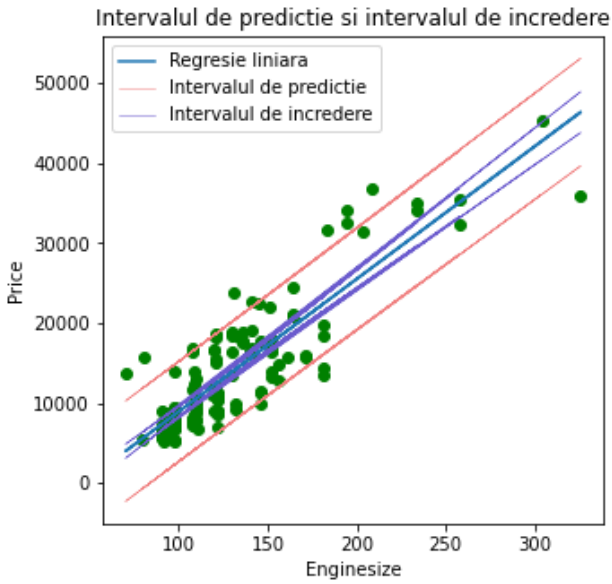

Figura 4.12: Intervalul de predicţie şi intervalul de încredere

Prin regresia lasso şi ridge s-au obţinut scoruri aproximativ egale cu scorul obţinut de regresia liniară, astfel că a fost reţinut pentru aplicare, modelul regresiei liniare.

4.2.5. Regresia random forest cu o variabilă explicativă

Prin aplicarea regresiei random forest la setul de date de antrenare a fost obţinut un scor de 0,9315, iar pentru datele de test s-a obţinut un scor de 0,9101, ambele superioare scorurilor

obţinute de regresia liniară simplă. Grafic regresia random forest se prezintă conform Figura 4.13:

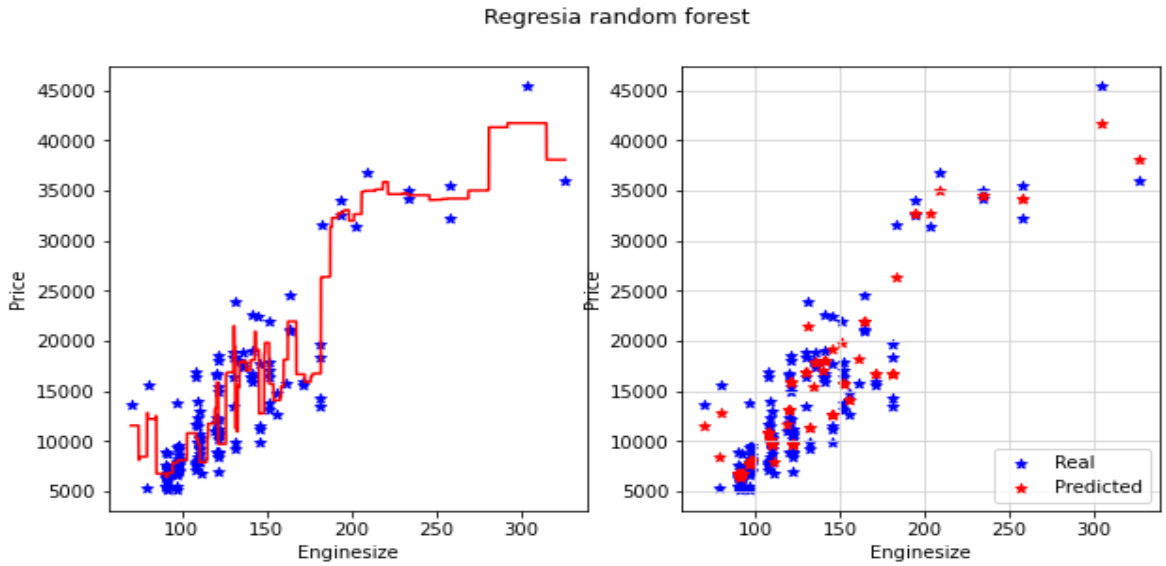

Figura 4.13: Regresia random forest simplă: enginesize → price

Regresia random forest este un model neliniar, astfel că se valorile estimate se prezintă sub forma unui nor de puncte, foarte apropiate de valorile reale. Având în vedere performanţele obţinute prin aplicarea modelului atât pe datele de antrenare cât şi pe datele de test, modelul a fost reţinut pentru aplicare.

4.2.6. Regresia liniară multiplă

În urma implementării modelului regresiei liniare multiple s-a obţinut **ecuaţia funcţiei liniare de regresie**:

$$y = -9524.689312612985 + [3.82247295 \;\; 82.78858615 \;\; 44.86893789 \;\; -71.26461143] * xi$$

Rezultatele modelării prin regresia liniară multiplă sunt prezentate în Figura 4.14:

OLS Regression Results

Dep. Variable:	price	R-squared:	0.828
Model:	OLS	Adj. R-squared:	0.823
Method:	Least Squares	F-statistic:	165.7
Date:	Sun, 23 May 2021	Prob (F-statistic):	1.20e-51
Time:	11:08:39	Log-Likelihood:	-1358.9
No. Observations:	143	AIC:	2728.
Df Residuals:	138	BIC:	2743.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-9524.6893	4973.971	-1.915	0.058	-1.94e+04	310.361
curbweight	3.8225	1.229	3.110	0.002	1.392	6.253
enginesize	82.7886	14.580	5.678	0.000	53.960	111.618
horsepower	44.8689	13.112	3.422	0.001	18.942	70.796
highwaympg	-71.2646	87.354	-0.816	0.416	-243.990	101.460

Omnibus:	5.568	Durbin-Watson:	1.988
Prob(Omnibus):	0.062	Jarque-Bera (JB):	6.419
Skew:	0.252	Prob(JB):	0.0404
Kurtosis:	3.907	Cond. No.	4.74e+04

Figura 4.14: Rezultate regresia liniară multiplă OLS

Ipotezele asupra proprietăţilor estimatorilor

Liniaritatea modelului: se poate observa vizual din graficul perechilor de variabile, existenţa unor relaţii dispuse liniar între fiecare variabilă independentă şi variabila dependentă.

Media erorilor este de -4,2231 fiind foarte apropiată de 0, ceea ce confirmă ipoteza.

Normalitatea erorilor se poate confirma vizual prin histograma erorilor prezentată în Figura 4.15 şi prin verificarea coeficientului de asimetrie (Skew) de 0,252 - apropiat de 0 şi a coeficientului de boltire (Kurtosis) de 3,907 - apropiat de 3. Putem considera că erorile reziduale sunt distribuite normal.

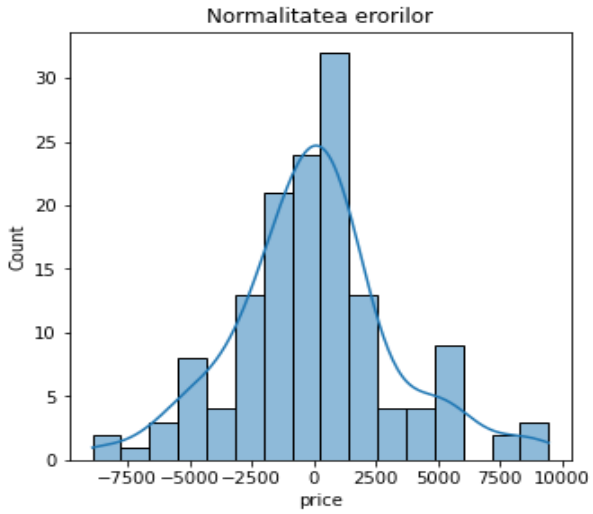

Figura 4.15: Normalitatea erorilor – regresia liniară multiplă

Homoscendasticitatea s-a verificat prin reprezentarea grafică a relaţiei dintre variabila x şi reziduuri, prezentată conform Figura 4.16:

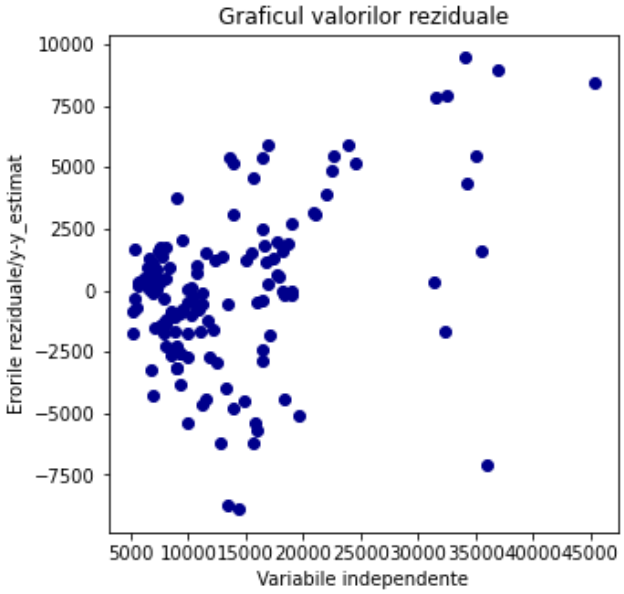

Figura 4.16: Graficul valorilor reziduale – regresia liniară multiplă

Se poate observa faptul că punctele nu formează un anumit model sau tipar, astfel că ipoteza homoscendasticităţii se confirmă.

Necorelarea erorilor a fost verificată prin testul Durbin Watson care are o valoare de 1,988. Pentru α=0,5 limita inferioară citită din tabelul Durbin Watson este de 1,694, iar limita superioară este 2,346. Întrucât valoarea testului se află în intervalul valorilor teoretice, putem considera ca nu există autocorelare între valorile reziduale.

Multicoliniaritatea

Din matricea de corelaţie între variabilele independente reiese faptul că există un nivel ridicat de corelaţie între toate variabilele independente implicate, respectiv coeficienţii de corelaţie în valoare absolută depăşesc 0,7.

De asemenea, factorul de inflamare a dispersiei erorilor (VIF) prezintă valori foarte ridicate:

```
          VIF   variables
0   59.072436  curbweight
1   45.814650  enginesize
2   23.913537  horsepower
3    8.786194  highwaympg
```

Având în vedere indicatorii de mai sus, este evident faptul că modelul este afectat de multicoliniaritate, care are următoarele efecte: coeficienţii de regresie vor fi instabili şi vor avea o eroare standard mare, intervalul de încredere al coeficienţilor de regresie va fi mare, coeficienţii de regresie vor fi greu de interpretat. Testarea semnificativităţii pantei de regresie şi a semnificativităţii globale va determina valori foarte mici ale testelor, astfel că este foarte greu de demonstrat respingerea ipotezei nule. În cazul multicoliniarităţii se recomandă eliminarea variabilelor independente care sunt foarte corelate între ele.

Cu toate acestea, am considerat că un model cu mai multe variabile va aduce mai multă informaţie şi este necesar pentru realizarea obiectivului studiului, acela de la întocmi un business plan pe baza rezultatelor obţinute, întrucât este imposibil de a realiza o strategie eficientă de business, care presupune întocmirea unui buget de venituri şi cheltuieli, şi a unui plan de producţie, pe baza informaţiei oferite de o singură variabilă. Considerăm că principalele variabile care influenţează în mod consistent preţul autovehiculelor sunt toate cele patru variabile independente: enginesize, curbweight, horsepower şi highwaympg. Faptul că cele patru variabile sunt corelate între ele arată că setul de date este corect întocmit şi informaţiile sunt reale, întrucât în realitate cele patru componente auto (în special capacitatea motorului, caii putere şi greutatea maşinii) sunt într-o anumită măsură corelate între ele şi au într-adevar un rol esenţial în determinarea preţului.

Prin urmare, dacă analizăm evoluţia coeficientului de determinare şi în paralel a coeficientului de determinare ajustat, putem observa următoarele:

- coeficientul de determinare R^2 înregistrează o creştere de la valoarea de 0,769 în cazul regresiei simple, la 0,828 în cazul regresiei multiple;

- coeficientul de determinare ajustat R^2_{adj} care în cazul regresiei simple era de 0,767 înregistrează o creştere la valoarea de 0,823 în cazul regresiei multiple.

Evoluţia lui R^2_{adj} pe măsură ce adaugăm variabile explicative este prezentată în Figura 4.17:

Variabile	Coeficientul de determinare ajustat
enginesize	0.769
enginesize + curbweight	0.801
enginesize + highwaympg	0.801
enginesize + horsepower	0.800
curbweight + horsepower + highwaympg	0.783
enginesize + curbweight + highwaympg	0.809
enginesize + horsepower + highwaympg	0.812
enginesize + curbweight + horsepower	0.823
enginesize + curbweight + horsepower + highwaympg	0.823

Figura 4.17: Evoluţia coeficientului de determinare ajustat

Coeficientul de determinare ajustat R^2_{adj} este corectat cu gradele de libertate, astfel că prin adăugarea de noi variabile inutile în model acesta va scade, iar prin adăugarea de variaible utile, acesta va creşte.

După cum se poate observa, pe măsură ce se adaugă variabile explicative în model, R^2_{adj} creşte, lucru care arată un câştig de informaţie ca urmare a adăugării de noi variabile.

Într-un model afectat de multicoliniaritate este greu de explicat modul în care variabila y este afectată de către fiecare variabilă independentă în parte, datorită instabilităţii coeficienţilor de regresie, dar multicoliniaritatea nu scade capacitatea predictivă a modelului (lucru care se va vedea în continuare), astfel că modelul va fi reţinut pentru aplicare.

4.2.7. Regresia random forest cu patru variabile explicative

Prin aplicarea regresiei random forest cu patru variabile explicative la setul de date de antrenare a fost obţinut un scor de 0,9852, iar pentru datele de test s-a obţinut un scor de 0,94, ambele superioare scorurilor obţinute de regresia liniară.

Pentru vizualizare grafică s-a modelat regresia random forest cu 2 variabile explicative, respectiv enginesize şi curbweight, care a obţinut un scor de 0,9798. Grafic regresia random forest cu doua variabile explicative se prezintă conform Figura 4.18:

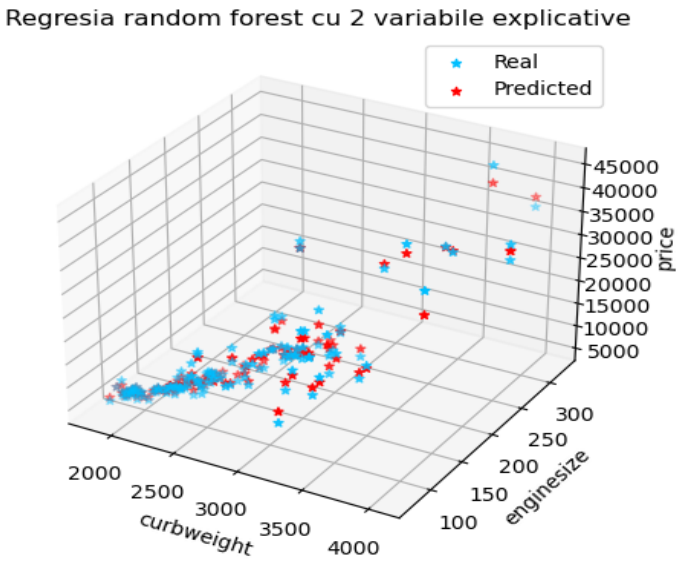

Figura 4.18: Scatterplot regresia random forest cu 2 variabile explicative

Regresia random forest este un model neliniar, astfel că se valorile estimate se prezintă sub forma unui nor de puncte, foarte apropiate de valorile reale. Având în vedere performanţele obţinute prin aplicarea modelului atât pe datele de antrenare cât şi pe datele de test, modelul a fost reţinut pentru aplicare.

4.2.8. Model alternativ bazat pe regresia random forest

Ca alternativă la modelele prezentate anterior a fost dezvoltat un model bazat pe regresiile random forest simple. Astfel, a fost modelată regresia random forest separat pentru fiecare variabilă independentă:

Regresia random forest enginesize → price a fost prezentată anterior.

Regresia random forest curbweight → price pentru setul de date de antrenare a obţinut un scor de 0,9559, iar pentru setul de date de test a obţinut un scor de 0,6999. Grafic modelul este prezentat în Figura 4.19:

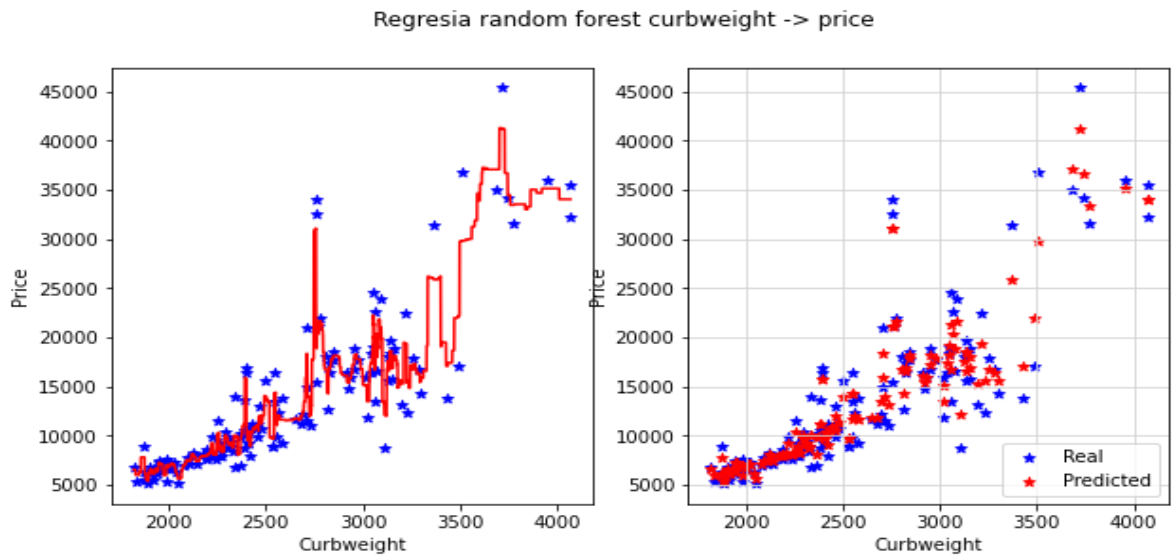

Figura 4.19: Regresia random forest simplă: curbweight → price

Regresia random forest horsepower → price pentru setul de date de antrenare a obţinut un scor de 0,9390, iar pentru setul de date de test a obţinut un scor de 0,8937. Grafic modelul este prezentat în Figura 4.20:

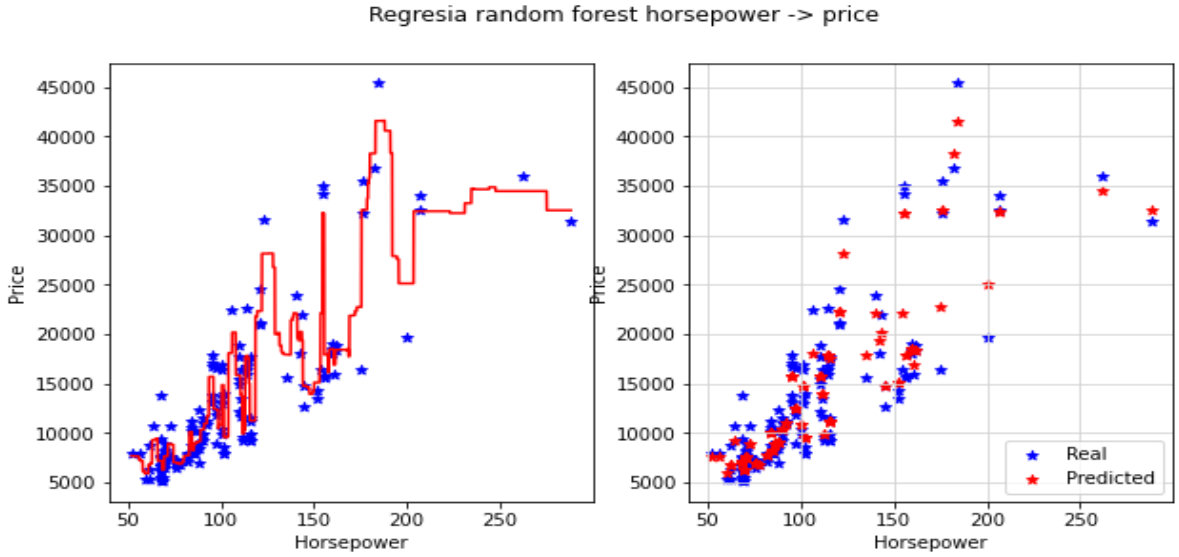

Figura 4.20: Regresia random forest simplă: horsepower → price

Regresia random forest highwaympg → price pentru setul de date de antrenare a obţinut un scor de 0,8283, iar pentru setul de date de test a obţinut un scor de 0,5966. Grafic modelul este prezentat în Figura 4.21:

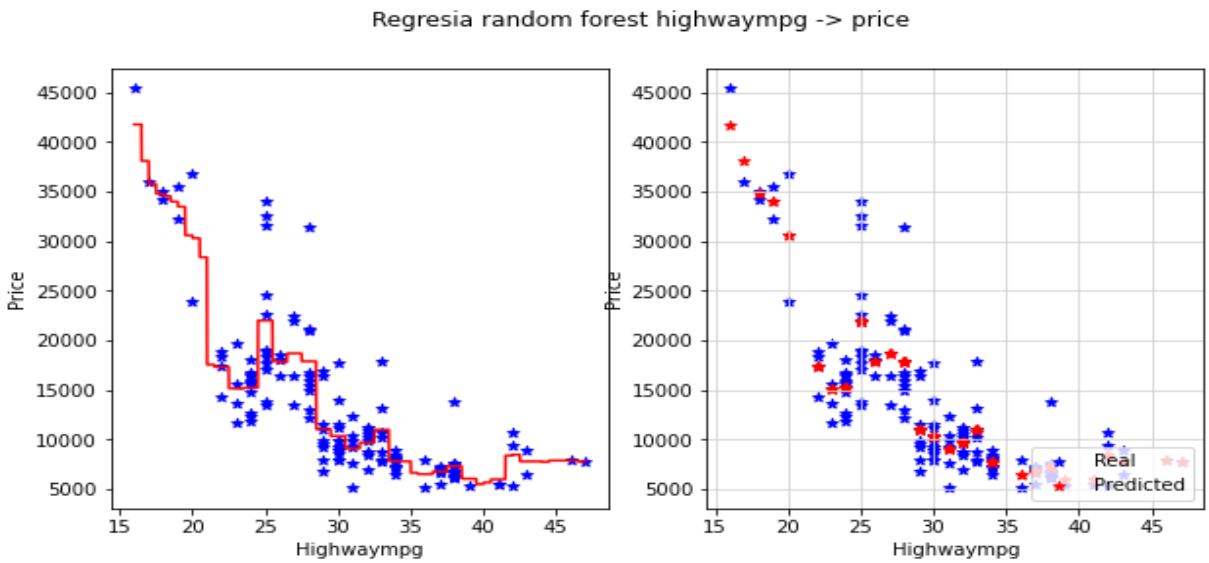

Figura 4.21: Regresia random forest simplă: highwaympg → price

Au fost dezvoltate patru metode de predicţie separate pentru fiecare model, iar pentru obţinerea rezultatului final s-a calculat media celor 4 răspunsuri individuale, medie ponderată cu scorul obţinut de fiecare metodă:

Pentru datele de antrenare s-a dezvoltat o metodă de predicţie bazată pe următoarea ecuaţie:

y_estimat = (y_estimat_enginesize * 0.9315 + y_estimat_horsepower * 0.9390 +
y_estimat_curbweight * 0.9559 + y_estimat_highwaympg * 0.8283)
/ (0.9315 + 0.9390 + 0.9559 + 0.8283)

Pentru datele de test s-a dezvoltat o metodă de predicţie bazată pe următoarea ecuaţie:

y_estimat = (y_estimat_enginesize * 0.9101 + y_estimat_horsepower * 0.8937 +
y_estimat_curbweight * 0.6999 + y_estimat_highwaympg * 0.5966)
/ (0.9101 + 0.8937 + 0.6999 + 0.5966)

4.2.9. Compararea rezultatelor modelelor dezvoltate

Pentru evaluarea performanţelor modelelor implementate s-au comparat p-normele vectorilor rezultaţi din diferenţele între variabila dependentă iniţiala y şi $\hat{y}$ estimat prin modelarea datelor, având în vedere faptul că variabila y şi variabila $\hat{y}$ estimat se prezintă sub forma unor vectori.

Dacă p este un număr real, iar $p \geq 1$, p-norma sau l_p-norma unui vector $x_i = (x_1, \dots, x_n)$ este [55]:

$$||x||_p \coloneqq (\sum_{i=1}^{n} |x_i|^p)^{1/p}$$

- dacă p = 1, avem 1-norma sau norma ll_1, iar distanţa care derivă din această normă se numeşte distanţa l_1; 1-norma este suma valorilor absolute ale coloanelor vectorului:

$$||\mathrm{x}||_1 \coloneqq \sum_{i=1}^{n} |x_i|$$

- dacă p = 2, avem norma euclidiană sau "radical din suma pătratelor"; astfel, pe un spaţiu euclidian n-dimensional $\boldsymbol{R}^n$, lungimea vectorului este dată de formula:

$$||\mathrm{x}||_2 \coloneqq \sqrt{x_1^2 + \cdots + x_n^2}$$

- dacă $p \to \infty$, p-norma se apropie de norma infinită sau norma maximă:

$$||\mathrm{x}||_\infty \coloneqq \max(|x_1|, \ldots, |x_n|)$$

Toate aceste norme sunt echivalente, întrucât toate definesc aceeaşi topologie [55].

Pentru evaluarea performanţelor modelelor implementate s-a verificat diferenţa sau distanţa între y iniţial şi $\hat{y}$ estimat. Pentru ca un model să fie eficient, trebuie ca distanţele între valorile iniţiale ale variabilei y şi valorile lui $\hat{y}$ estimat să fie cât mai mici.

Astfel, dacă avem:

$$d_{l_p}(y_i, \hat{y}_i) = \sqrt[p]{\sum |y_i - \hat{y}_i|^{\mathrm{p}}}$$

- dacă p = 1, avem:

$$d_{l_1} = \sum |y_i - \hat{y}_i|, \quad \text{iar} \quad \frac{1}{n} * d_{l_1} = \frac{1}{n} * \sum |y_i - \hat{y}_i| = \mathrm{MAE}$$

- dacă p = 2, avem:

$$d_{l_2} = \sqrt{\sum (y_i - \hat{y}_i)^2}, \text{ iar } \sqrt{\frac{1}{n}} * d_{l_2} = \sqrt{\frac{1}{n}} * \sqrt{SSE} = RMSE$$

- dacă $p \to \infty$, avem valoarea maximă a vectorului, care arată care este valoarea lui $\hat{y}$ estimat, cea mai depărtată de y iniţial:

$$d_{l_\infty} = \max\{(y_i - \hat{y}_i), i = \overline{1, n}\}$$

Au fost calculaţi pentru fiecare dintre modele indicatorii, iar situaţia este prezentată în Figura 4.22:

Evaluarea performantelor		Reg liniara simpla	Random forest 1 variabila explicativa	Reg liniara multipla	Random forest 4 variabile explicative	Model alternativ
R^2	train	0.77	0.93	0.83	0.99	-
	test	0.75	0.91	0.78	0.94	-
$d_{l_1}(MAE)$	train	2,814.75	1,594.25	2,352.26	639.18	1,201.17
	test	2,960.71	1,801.56	2,777.90	1,455.20	1,902.35
$d_{l_2}(RMSE)$	train	3,754.74	2,043.74	3,241.21	950.12	1,640.88
	test	4,132.71	2,496.18	3,922.16	2,008.81	2,792.89
$d_{l_\infty}(max)$	train	10,394.15	6,105.32	9,450.71	4,089.72	5,278.72
	test	14,315.78	6,435.13	14,018.59	6,706.10	10,807.45

Figura 4.22: Tabel compararea rezultatelor modelelor

Modelul random forest cu 4 variabile explicative a obţinut cel mai mare coeficient de determinare R^2, respectiv 0.99 pentru datele de antrenare şi 0.94 pentru datele de test.

Cele mai mici valoari ale MAE şi $RMSE$ atât pentru datele de antrenare cât şi pentru datele de test au fost obţinute de modelul random forest cu 4 variabile explicative. După random forest cu 4 variabile explicative, cele mai mici valori ale MAE şi $RMSE$ pentru datele de antrenare au fost obţinute de modelul alternativ, iar pentru datele de test, de către random forest cu o variabilă explicativă, diferenţa pentru datele de test între cele două fiind foarte mică.

În ceea ce priveşte valoarea maximă a vectorului rezultat din diferenţa între vectorul y initial şi vectorul $\hat{y}$ estimat, pe datele de antrenare cea mai mică valoare a fost obţinută de către modelul random forest cu o variabilă explicativă, imediat urmând modelul alternativ, iar pe datele de test cea mai mică valoare a fost obţinută de random forest cu o variabilă explicativă, la diferenţă foarte mică fiind modelul random forest cu 4 variabile explicative.

S-au acordat punctaje modelelor în funcţie de performanţa totală, performanţa obţinută pe datele de antrenare şi performanţa pe datele de test, iar rezultatele sunt prezentate în Figura 4.23:

Model	Punctaj pe datele de antrenare	Punctaj pe datele de test	Punctaj date antrenare şi test
Random forest 4 variabile explicative	9	8	17
Random forest 1 variabilă explicativă	3	7	10
Model alternativ	6	3	9

Figura 4.23: Punctaj comparativ modele implementate

În concluzie, **cel mai performant model este regresia random forest cu 4 variabile** explicative, care a obţinut cele mai bune rezultate atât pe datele de antrenare cât şi pe datele de test, obţinând astfel şi un raport optim între bias şi varianţă. Chiar dacă modelul este afectat de multicoliniaritate, aceasta nu scade capacitatea sa predictivă, care este cea mai ridicată dintre toate modelele aplicate.

Al doilea cel mai eficient model pe datele de antrenare este **modelul alternativ**, iar pe datele de test **modelul regresiei random forest simple** cu o variabilă explicativă. Aceasta arată faptul că modelul alternativ are un bias mai mic decât regresia random forest simplă şi o variaţie mai mare decât aceasta, în timp ce regresia random forest simplă are un bias mai mare şi o variaţie mai mică decât modelul alternativ.

Prin urmare, dacă nu ne interesează să extrapolăm modelul pentru alte date decât cele pe care le avem la dispoziţie, modelul alternativ are performanţe mai ridicate decât modelul random forest simplu. Piaţa auto americană are un anumit specific care e posibil să nu mai fie întalnit în altă parte, fiind caracterizată de preferinţa pentru autovehiule de dimensiuni mari, cu motoare de capacitate mare, existând o corelaţie ridicată între greutatea autovehiculului (curbweight), capacitatea cilindrică (enginesize) şi randamentul motorului (horsepower), lucru care se observă din datele care stau la baza studiului. Piaţa auto din Europa de exemplu este diferită, fiind caracterizată de preferinţa pentru autovehicule de dimensiuni mai mici, care nu necesită neaparat motoare de capacitate mare, iar corelaţia între principalele variabile explicative din prezentul studiu este foarte posibil să nu fie aşa de ridicată ca în cazul pieţei auto americane.

Mai mult, întrucât modelul alternativ ţine cont de toate cele patru variabile importante în explicarea modelului este mai adecvat pentru scopul studiului, acela de a determina care sunt variabilele importante care influenţează preţul autivehiculelor şi de a întocmi un business plan eficient.

Următorul model ca şi eficienţă este regresia liniară multiplă, care depăşeşte performanţele regresiei liniare simple, arătând din nou faptul că multicoliniaritatea nu reduce capacitatea predictivă a modelului.

Regresia random forest nu poate extrapola rezultatele pentru valori mai mari decât limita maximă sau mai mici decât limita minimă a valorilor variabilelor explicative.

$$x \geq x_{min} \text{ şi } x \leq x_{max}$$

În afara intervalului de valori ale variabilelor explicative regresia random forest nu poate da rezultate. Spre deosebire de regresia random forest, modelul regresiei liniare funcţionează şi

pentru valori din afara intervalului de valori ale variabilelor independente. Prin urmare, dacă se doreşte estimarea variabilei dependente şi pentru valori dincolo de limita minimă sau maximă a variabilelor explicative, se poate utiliza modelul regresiei liniare.

Toate cele cinci modelele sunt valide şi cu performanţe bune, fiecare putând fi utilizat pentru estimarea preţului unui autovehicul pe baza setului de variabile explicative. Cel mai performant model, care oferă cea mai multă informaţie despre variabila dependentă preţ şi care răspunde cel mai bine scopului prezentului studiu este regresia random forest cu patru variabile explicative (enginesize, curbweight, horsepower şi highwaympg), întrucât vectorul de valori estimate este la cea mai mică distanţă de vectorul de valori iniţiale şi reuşeşte să obţină cel mai optim compromis între bias-ul pe datele de antrenare şi varianţa pe datele de test, putând fi extrapolat şi la alte seturi de date.

Bibliografie

[1] Wikipedia, https://ro.wikipedia.org/wiki/Econometrie (accesare: 02.04.2021).

[2] Wikipedia, https://ro.wikipedia.org/wiki/Analiza_de_regresie (accesare: 02.04.2021).

[3] D. Cielen, A. D. B. Meysman and M. Ali, Introducing Data Science, Shelter Island: Manning Publications Co., 2016.

[4] C. H. Lau, https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492 (accesare: 05.04.2021).

[5] M. Mayo, https://www.kdnuggets.com/2019/06/7-steps-mastering-data-preparation-python.html (accesare: 03.03.2021).

[6] A. Bhandari, https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/ (accesare: 03.03.2021).

[7] Official website IBM, https://www.ibm.com/cloud/learn/exploratory-data-analysis (accesare: 03.03.2021).

[8] L. Sasu, https://www.slideshare.net/lmsasu/curs-2-data-mining (accesare: 04.03.2021).

[9] P. Sharma, https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/ (accesare: 04.03.2021).

[10] Aman1608, https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/ (accesare: 05.03.2021).

[11] Wikipedia, https://en.wikipedia.org/wiki/Machine_learning (accesare: 05.03.2021).

[12] Wikipedia, https://ro.wikipedia.org/wiki/Statistic%C4%83 (accesare: 07.03.2021).

[13] D. Danciulescu, http://inf.ucv.ro/documents/danciulescu/curs4-curs-5-curs6.pdf (accesare: 20.11.2020).

[14] D. Maniu, http://www.phys.ubbcluj.ro/~dana.maniu/BIOSTAT/C2.pdf (accesare: 20.11.2021).

[15] UMFVC, http://www.umfcv.ro/files/b/i/Biostatistica%20MG%20-%20Cursul%205%20-%20Corelatii.pdf (accesare: 20.11.2020).

[16] Wikipedia, https://ro.wikipedia.org/wiki/Covarian%C8%9B%C4%83 (accesare: 20.11.2020).

[17] Slideshare, https://www.slideshare.net/Cattta89/regresie (accesare: 01.12.2020).

[18] M. Chavent, http://www.math.u-bordeaux.fr/~mchave100p/wordpress/wp-content/uploads/2013/10/ModStat_C1_pres.pdf (accesare: 20.02.2021).

[19] R. Rakotomalala, Econometrie La regression lineaire simple et multiple, Lyon: Universite Lumiere Lyon 2, 2018.

[20] M. T. Coada, https://www.slideshare.net/tiberiumarian92/49855810-capitolul2regresialiniarapp133slideej (accesare: 25.11.2020).

[21] A. Fahad, https://machinelearningmind.com/2019/10/27/assumptions-of-linear-regression-how-to-validate-and-fix/ (accesare: 04.04.2021).

[22] V. Cristescu și T. Sâia, http://www.imst.pub.ro/Upload/Sesiune/ComunicariStiintifice/Lucrari_2015/06.16/16_L36.pdf (accesare: 07.04.2021).

[23] R. Atha, https://medium.com/swlh/multi-linear-regression-using-python-44bd0d10082d (accesare: 07.04.2021).

[24] Wikipedia, https://ro.xcv.wiki/wiki/Robust_regression (accesare: 21.03.2021).

[25] Wikipedia, https://en.wikipedia.org/wiki/Random_sample_consensus#:~:text=Random%20sample%20consensus%20(RANSAC)%20is,as%20an%20outlier%20detection%20method. (accesare: 05.02.2021).

[26] Official website scikit-learn, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RANSACRegressor.html (accesare: 15.03.2021).

[27] D. Zaharie, https://staff.fmi.uvt.ro/~daniela.zaharie/dm2017/RO/curs/dm2017_curs11.pdf (accesare: 31.03.2021).

[28] A. Chakure, https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f (accesare: 01.04.2021).

[29] J. Starmer, https://www.youtube.com/user/joshstarmer (accesare: 01.04.2021).

[30] D. Mwiti, https://neptune.ai/blog/random-forest-regression-when-does-it-fail-and-why (accesare: 03.04.2021).

[31] S. Glen, https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/adjusted-r2/ (accesare: 31.03.2021).

[32] A. Chugh, https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e (accesare: 04.09.2021).

[33] J. Brownlee, https://machinelearningmastery.com/regression-metrics-for-machine-learning/ (accesare: 08.04.2021).

[34] D. N. Sadawi, https://www.youtube.com/user/DrNoureddinSadawi/playlists (accesare: 01.12.2020).

[35] The Pennsylvania State University website, https://online.stat.psu.edu/stat501/lesson/6/6.2 (accesare: 19.03.2021).

[36] The Pennsylvania State University website, https://online.stat.psu.edu/stat462/node/135/ (accesare: 19.03.2021).

[37] A. Birlutiu, http://adrianabirlutiu.uab.ro/cursuri/MIRF/note_curs_lab_5.pdf (accesare: 30.03.2021).

[38] A. Birlutiu, http://adrianabirlutiu.uab.ro/cursuri/MIRF/2018curs5.pdf (accesare: 15.03.2021).

[39] S. Singh, https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229 (accesare: 28.02.2021).

[40] Wikipedia, https://ro.xcv.wiki/wiki/Bias%E2%80%93variance_tradeoff (accesare: 28.02.2021).

[41] D. Patel, https://www.youtube.com/c/codebasics/playlists (accesare: 15.11.2020).

[42] Wikipedia, https://en.wikipedia.org/wiki/Python_(programming_language) (accesare: 01.02.2021).

[43] Official website Python, https://www.python.org/ (accesare: 01.02.2021).

[44] Official website Python Package Index, https://pypi.org/ (accesare: 01.02.2021).

[45] Wikipedia, https://en.wikipedia.org/wiki/Project_Jupyter (accesare: 01.02.2021).

[46] Official website Project Jupyter, https://jupyter.org/ (accesare: 01.02.2021).

[47] Wikipedia, https://en.wikipedia.org/wiki/NumPy (accesare: 02.02.2021).

[48] Official website Numpy, https://numpy.org/ (accesare: 02.02.2021).

[49] Wikipedia, https://en.wikipedia.org/wiki/Pandas_(software) (accesare: 02.02.2021).

[50] Official website Pandas, https://pandas.pydata.org/ (accesare: 02.02.2021).

[51] Wikipedia, https://en.wikipedia.org/wiki/Matplotlib (accesare: 03.02.2021).

[52] Official website Matplotlib, https://matplotlib.org/ (accesare: 03.02.2021).

[53] Wikipedia, https://en.wikipedia.org/wiki/Scikit-learn (accesare: 04.02.2021).

[54] Official website Scikit-learn, https://scikit-learn.org/stable/ (accesare: 03.02.2021).

[55] Wikipedia, https://ro.wikipedia.org/wiki/Norm%C4%83_(matematic%C4%83) (accesare: 25.05.2021).