Electric Vehicle Population Prediction

Studiu pentru prezicerea de informatii

Student: *Hosu Razvan*

Iunie 2025

Cuprins

1 Introducere și motivația alegerii bazei de date **2**

2 Contextul sursei de date și al proiectului, cerințe, ce dorim să obținem **3**

3 Aspecte teoretice **5**

4 Implementarea aspectelor teoretice în cadrul proiectului **7**
4.1 Încărcarea datasetului și analiza structurii 7
4.2 Definirea scopului analizei 7
4.3 Detectarea valorilor lipsă și a duplicatelor 7
4.4 Imputarea sau eliminarea valorilor lipsă 8
4.5 Curățarea datelor 8
4.6 Analiza vizuală a datelor 9
4.7 Matricea de corelație și interpretarea ei 9
4.8 Selecția caracteristicilor: entropie, Information Gain și Gini Index 10
4.9 Evaluarea și compararea modelelor 11
4.10 Predicția pentru o intrare nouă 12
4.11 Importanța caracteristicilor și concluzii 12

5 Testare și validare **13**
5.1 Metodologia generală de testare 13
5.2 Modele de regresie analizate 13
5.3 Validare încrucișată 14
5.4 Rezultate și comparații 14
5.5 Alegerea modelului final 15
5.6 Limitări și direcții viitoare 15

6 Rezultate **16**
6.1 Explorare inițială a datelor 16
6.2 Curățarea datelor 16
6.3 Agregare spațio-temporală 16
6.4 Construirea și performanța modelelor de regresie 16
6.5 Validarea modelelor de predicție 17
6.6 Aplicarea modelului pe o instanță nouă 17
6.7 Analiza sensibilității (extensii viitoare) 18
6.8 Interpretarea și raportarea rezultatelor 18

7 Concluzii **19**

8 Referințe **21**

1 Introducere și motivația alegerii bazei de date

Tranziția către un transport sustenabil reprezintă una dintre cele mai mari provocări ale societății contemporane, atât din punct de vedere tehnologic, cât și din punct de vedere economic, social și politic. În acest context, vehiculele electrice (Electric Vehicles – EV) joacă un rol central în decarbonizarea sectorului transporturilor și în reducerea dependenței de combustibili fosili.

Adoptarea vehiculelor electrice a cunoscut o creștere semnificativă în ultimul deceniu, alimentată de progresele în tehnologia bateriilor, politicile de sprijin guvernamental, dar și de schimbarea comportamentului consumatorilor. Cu toate acestea, penetrarea pieței nu este uniformă, iar distribuția geografică, tipologia vehiculelor și preferințele utilizatorilor diferă considerabil între regiuni. Înțelegerea acestor diferențe este esențială pentru proiectarea unor politici publice eficiente, alocarea optimă a resurselor pentru infrastructura de încărcare, precum și pentru încurajarea cercetării și dezvoltării în domeniul mobilității electrice.

Lucrarea de față își propune analiza unui set de date reale privind vehiculele electrice înmatriculate într-o anumită regiune, folosind instrumente moderne de analiză a datelor în mediul *Jupyter Notebook*. Alegerea acestei platforme se datorează flexibilității oferite în explorarea, preprocesarea și vizualizarea datelor, fiind un standard de facto în cercetarea științifică din domeniul științei datelor.

Motivația alegerii temei

Subiectul este unul de actualitate, plasat la intersecția dintre sustenabilitate, tehnologie și politici publice. Într-o lume în care orașele sunt din ce în ce mai aglomerate și calitatea aerului este o preocupare majoră, vehiculele electrice oferă o alternativă viabilă la transportul convențional. Astfel, analiza distribuției și caracteristicilor acestora nu are doar un interes teoretic, ci și o utilitate practică imediată.

De asemenea, dezvoltarea accelerată a pieței EV ridică noi întrebări privind infrastructura, comportamentul consumatorilor și viitorul industriei auto. Prin analiza datelor, putem extrage tipare relevante, putem identifica zone cu potențial ridicat pentru investiții în stații de încărcare și putem evidenția direcții de dezvoltare.

Motivația alegerii bazei de date

Setul de date utilizat a fost selectat datorită următoarelor considerente:

- Este un set de date **public**, provenit de la o **autoritate guvernamentală**, ceea ce garantează acuratețea și actualitatea acestuia;
- Conține o gamă variată de **atribute relevante**, precum marca vehiculului, tipul de propulsie, tipul caroseriei, codul ZIP al înmatriculării și anul înmatriculării;
- Permite efectuarea unei **analize exploratorii detaliate**, fiind structurat și ușor de procesat cu biblioteci Python precum `pandas`, `matplotlib` și `seaborn`;
- Datele sunt **suficient de numeroase și diversificate** pentru a genera rezultate semnificative statistic și a realiza vizualizări expresive;
- Este un set de date care poate fi **replicat** și utilizat de alți cercetători pentru validare sau extindere.

Prin intermediul acestei lucrări, ne propunem să realizăm o explorare sistematică a setului de date, să generăm vizualizări relevante, să extragem cunoștințe utile și să evaluăm dacă tendințele observate corespund ipotezelor actuale din literatura de specialitate. De asemenea, prin prezentarea metodei de lucru și a instrumentelor folosite, dorim să oferim un exemplu de bune practici în prelucrarea datelor deschise cu aplicații în domeniul transporturilor.

2 Contextul sursei de date și al proiectului, cerințe, ce dorim să obținem

2.1 Sursa și caracteristicile setului de date

Setul de date *Electric Vehicle Population* a fost preluat de pe platforma Kaggle (`https://www.kaggle.com/datasets/willianoliveiragibin/electric-vehicle-population`), încărcat de Willian Oliveira Gibin și actualizat ultima dată acum doi ani. Structurat într-un singur fișier CSV de aproximativ 34 MB, dataset-ul conține peste 150 000 de înregistrări detaliate despre vehiculele electrice înmatriculate în diverse regiuni, pentru perioadele 2010–2022. Fiecare înregistrare include următoarele atribute cheie:

- **Marca și modelul vehiculului** – pentru a surprinde diversitatea producătorilor şi a tipurilor comerciale;
- **Tipul de propulsie** – baterie electrică pură (BEV) sau hibrid plug-in (PHEV);
- **Anul înmatriculării** – necesar pentru analiza evoluției în timp;
- **Codul poștal (ZIP)** – pentru distribuția geografică la nivel local şi regional;
- **Alte atribute secundare** – capacitatea bateriei, autonomia oficială, tipul de încărcare, etc.

Gradul de *usability* al dataset-ului, evaluat la 10/10, se datorează documentației complete, structurii coerente şi calității metadatelor, ceea ce permite reproducerea ușoară a analizei.

2.2 Obiectivele proiectului

Prin această lucrare ne propunem:

1. **Explorarea inițială (EDA)** – descrierea statistică a variabilelor, identificarea valorilor lipsă, a outlier-ilor şi a relațiilor inițiale între atribute.
2. **Curățarea datelor** – imputarea valorilor lipsă, eliminarea înregistrărilor corupte, transformări (de exemplu, conversia datelor de tip text în categorii numerice).
3. **Agregare temporală și geografică** – gruparea înmatriculărilor pe ani și pe regiuni (coduri ZIP sau județe), pentru a evidenția tendințele spațio-temporale.
4. **Construirea modelului predictiv** – utilizarea regresiei liniare simple și multiple pentru estimarea numărului total de înmatriculări EV în anul 2024, pe baza datelor istorice (metoda OLS din scikit-learn).

5. **Validarea modelului și estimarea erorii** – aplicarea tehnicii de cross-validation (k-fold cu $k = 5$), calculul metricilor MSE, RMSE și R^2, precum și construirea intervalelor de încredere şi a distribuției reziduurilor.

6. **Analiza sensibilității** – simulări Monte Carlo pe parametrii cheie (de exemplu, rata de creștere anuală medie) pentru a cuantifica impactul variațiilor externe (politici guvernamentale, evoluții tehnologice, factori economici).

7. **Interpretarea și raportarea rezultatelor** – identificarea factorilor care influențează cel mai puternic adoptarea EV-urilor şi formularea unor recomandări pentru factorii de decizie.

2.3 Considerații privind predicția și marja de eroare

Deși predicțiile sunt ancorate în tendințele istorice, următoarele elemente pot modifica semnificativ evoluția reală a înmatriculărilor:

- **Politici publice și subvenții** – creșterea sau reducerea stimulentelor financiare pentru EV pot accelera sau frâna adoptarea;
- **Inovații tehnologice** – progresele în densitatea energetică a bateriilor, infrastructura de încărcare rapidă și costurile de producție;
- **Fluctuații economice** – variabile macroeconomice (PIB, inflație) și prețurile la energia electrică;
- **Preferințe de consum** – schimbări culturale și reglementări legate de emisii.

Pentru a reflecta aceste incertitudini, modelul va raporta:

- *Intervale de încredere 95%* pentru fiecare predicție anuală;
- *Marja de eroare maximă* tolerată (de ex. $\pm 5\%$ față de valoarea estimată), în conformitate cu standardele de validare statistică;
- *Rezultate ale simulărilor Monte Carlo*, care vor ilustra distribuția probabilistică a predicției sub scenarii alternative.

2.4 Relevanța și aplicabilitatea rezultatelor

Predicțiile și concluziile desprinse din acest proiect pot fi utilizate pentru:

- **Planificarea și extinderea infrastructurii de încărcare**, prin identificarea regiunilor cu potențial de creștere rapidă;
- **Elaborarea politicilor publice**, prin fundamentarea deciziilor privind subvențiile şi taxele aferente EV-urilor;
- **Strategii comerciale pentru producătorii auto**, prin adaptarea ofertei la cererea regională şi segmentarea pieței pe categorii de utilizatori;
- **Studii academice ulterioare**, prin reutilizarea setului de date și a codului open-source publicat în notebook.

3 Aspecte teoretice

Studiile recente subliniază complexitatea factorilor care influențează adopția vehiculelor electrice (EV). De exemplu, Coffman et al. (2017) revizuiesc factorii socio-economici ce afectează adoptarea EV, iar Hardman et al. (2018) analizează preferințele consumatorilor și interacțiunile acestora cu infrastructura de încărcare. În mod similar, Axsen et al. (2015) evidențiază heterogenitatea stilurilor de viață ale potențialilor cumpărători de EV plug-in. Analize mai noi, folosind tehnici de Machine Learning, caută să modeleze direct înmatriculările EV. De exemplu, Devarasan et al. (2025) realizează o analiză a dinamicii pieței EV din India în intervalul 2014–2024 cu ajutorul algoritmilor ML. Într-un studiu global recent, Yeh și Wang (2023) propun un model de predicție a vânzărilor de EV bazat pe ML, confirmând că modelele ML atinge niveluri înalte de acuratețe în prognoza EV. Aceste lucrări poziționează cercetarea noastră în contextul "state-of-the-art" pentru analize predictive de EV.

În ceea ce privește algoritmii de regresie utilizați, clasificăm trei modele importante:

Random Forest Regressor. Acest model este o pădure de arbori decizionali construită prin agregarea rezultatelor unor arbori diferențiați prin eşantionare bootstrap și selecție aleatorie a caracteristicilor. Teoretic, Random Forest (RF) reduce semnificativ varianța față de un arbore unic, datorită votului mediu a numeroşi arbori generaţi independent. Avantajele RF includ capacitatea de a învăța relații complexe non-liniare, rezistența la supraînvățare („overfitting") și obținerea de măsuri interne de importanță a caracteristicilor. Totodată, dezavantajele constau în complexitate computațională crescută și interpretabilitate redusă (rezultatele nu sunt ușor de tradus în reguli explicite). În contextul datelor de înmatriculări EV (unde pot exista variabile mixte și efecte complexe), RF este potrivit deoarece poate modela interacțiuni multiple fără a necesita specificarea prealabilă a formei relațiilor. Conform lui Breiman (2001), acest model atinge de obicei performanțe superioare celor ale unui arbore individual în predicție.

Decision Tree Regressor. Algoritmul de regresie bazat pe arbori de decizie construiește un arbore de tip CART (Classification and Regression Tree) prin împărțirea recursivă a setului de date pe baza unor condiții de tip prag pe caracteristici continue sau discrete. Quinlan (1986) descrie fondamentele acestor arbori, subliniind că ei oferă un model transparent (sub formă de reguli "if-then") și pot captura ușor relații nenelineare. Principalele avantaje ale arborelui de decizie unic sunt simplitatea de interpretare și viteza relativă de antrenare. Dezavantajele majore sunt variabilitatea mare și predispoziția la supraînvățare: un arbore neprunin poate să fie prea adaptat zgomotului din date, precum și sensibilitatea la mici modificări ale setului de date. În plus, arborele de decizie poate favoriza caracteristici cu multe nivele (posibil sesizând variații irelevante). În analiza noastră, arborii de decizie pot identifica rapid tiparele dominante în datele de înmatriculare (de ex. segmente de vehicule) dar trebuie controlați prin tehnici de poduire („pruning") sau reglarea adâncimii pentru a evita erori mari de predicție pe date noi.

K-Nearest Neighbors Regressor. Algoritmul k-NN este un model non-parametric care, în faza de inferență, estimează valoarea țintă pentru o observație necunoscută ca media valorilor celor k vectori de antrenament cei mai apropiați, după o metrică (de obicei distanța Euclidiană). Cover și Hart (1967) au demonstrat că regula 1-NN este robustă din punct de vedere teoretic, având rata de eroare într-o anumită limită față de cel mai bun model (Bayes). Avantajele KNN includ simplitatea și faptul că nu presupune o formă funcțională predefinită (nu necesită antrenament explicit). Cu toate acestea, dezavantajele sale sunt importante: modelul este costisitor din punct de vedere al stocării și al timpului de evaluare (deoarece trebuie parcurs tot setul de antrenament), iar performanța sa scade dramatic în spații de dimensiune mare („curse of dimensionality"). În plus, KNN este foarte sensibil la normalizarea caracteristicilor și la prezența outlier-ilor. În setul de date EV, unde există numeroase atribute (ex. regiune, an fabricație, caracteristici tehnice ale vehiculului), KNN ar putea fi ineficient deoarece fiecare caracteristică suplimentară diluează relevanța distanței. Totuși, KNN poate servi ca referință simplă de comparație în absența unor ipoteze despre distribuție.

Preprocesare și selecție de caracteristici. Înainte de antrenarea modelelor, se efectuează analiza exploratorie a datelor (EDA) pentru a înțelege distribuțiile variabilelor, a detecta corelații sau valori aberante și pentru a decide normalizări adecvate. Operațiile de curățare a datelor (eliminarea valorilor lipsă sau eronate, filtrarea outlier-ilor) sunt critice pentru a îmbunătăți calitatea predicțiilor. După preprocesare, se aplică selecția caracteristicilor: un exemplu de metodă este utilizarea scorului de corelație Pearson între fiecare caracteristică și variabila țintă, pentru a filtra (în mod statistic) variabilele cele mai relevante. Guyon și Elisseeff (2003) discută astfel de metode de filtrare a caracteristicilor pe baza scorurilor de corelație sau al altor criterii statistice. Acest pas reduce dimensiunea spațiului de intrare și poate îmbunătăți generalizarea modelelor. În fine, setul de date se împarte în subseturi de antrenament și de test (de obicei 70–30% sau 80–20%) pentru a evalua corect performanța predictivă pe date nevăzute, procedeu standard în ML.

Evaluare și comparație. Performanța modelelor este măsurată prin indicatori de eroare și de acuratețe, precum eroarea pătratică medie (MSE) și coeficientul de determinare, notat R^2. MSE penalizează sever erorile mari, în timp ce R^2 indică proporția variației explicate de model. Deși MSE este utilizat frecvent, Willmott și Matsuura (2005) arată că pentru interpretare, eroarea absolută medie (MAE) poate fi adesea mai intuitivă decât rădăcina MSE (totuși, în practica ML, R^2 și MSE rămân standard). În cazul nostru ipotetic, rezultatele tipice sugerează că Random Forest obține cele mai bune scoruri (MSE mai mic, R^2 mai mare) comparativ cu un arbore simplu sau KNN, datorită stabilității și capacității sale de agregare. Aceste observații corespund literaturii analogice; de pildă, în previziuni de vânzări, RF a depășit semnificativ performanța KNN și a arborilor decizionali individuali.

4 Implementarea aspectelor teoretice în cadrul proiectului

4.1 Încărcarea datasetului și analiza structurii

Primul pas a constat în încărcarea setului de date despre vehicule electrice în mediu Python (Jupyter). Am utilizat biblioteca `pandas` pentru a citi fișierul CSV:

```
df = pd.read_csv("Electric_Vehicle_Data.csv")
print(df.head())
```

Analiza inițială a structurii datasetului a fost realizată prin metode precum `df.head()`, `df.info()` și `df.describe()`. Acestea ne-au permis să inspectăm primele înregistrări, tipurile de date ale coloanelor (numeric vs. categorial) și statisticile descriptive (număr de valori non-nule, media, deviația standard etc.). Din analiza descriptivă am observat că datasetul conține aproximativ $150,000$ de înregistrări, cu coloane precum *VIN*, *Make*, *Model*, *Model Year*, *Electric Vehicle Type*, *Electric Range*, *Base MSRP* și altele. Am verificat tipurile de date: de exemplu, coloanele *Model Year* și *Electric Range* sunt numerice, în timp ce *Make*, *Model* și *Electric Vehicle Type* sunt categorice. Această explorare preliminară a datelor ne-a pregătit pentru pașii de preprocesare ulterioară și a confirmat particularitățile teoretice privind diferențierea datelor numerice/categorice.

4.2 Definirea scopului analizei

Scopul principal al analizei a fost identificarea unei metode de predicție eficiente și extragerea de cunoștințe relevante din date. Mai precis, ne-am propus să construim un model predictiv pentru autonomia electrică (`Electric Range`) a vehiculelor, valorificând tehnicile de învățare automată discutate în capitolul teoretic anterior. Astfel, obiectivul a fost dublu: (1) să prezicem cât mai precis o variabilă țintă numerică folosind algoritmi de regresie (așa cum am abordat în capitolul despre regresie), și (2) să înțelegem care variabile (caracteristici) influențează cel mai mult acea predicție (aspectele de interpretabilitate și selecție de caracteristici descrise anterior). Acest pas de „formulare a sarcinii” conectează teoria cu practica: de exemplu, dacă în capitolul precedent am vorbit despre criterii de selecție a caracteristicilor, atunci aici vom aplica aceste criterii asupra datasetului real.

4.3 Detectarea valorilor lipsă și a duplicatelor

Pentru a asigura calitatea datelor, am identificat întâi valorile lipsă și duplicatele. Cu `df.isnull().sum()` am calculat numărul de valori nule pe fiecare coloană și am construit un tabel de tip:

```
missing_values = df.isnull().sum()
missing_percentage = (missing_values / len(df)) * 100
missing_data = pd.DataFrame({
"Missing Count": missing_values,
"Percentage (%)": missing_percentage
})
missing_data = missing_data[missing_data["Missing Count"] > 0]
print(missing_data)
```

Rezultatul a arătat că majoritatea coloanelor aveau foarte puține valori lipsă (de exemplu doar 3 în *County*, *City* etc.), însă coloana *Legislative District* prezenta un număr semnificativ de valori lipsă (aprox. 341 din $150,000$, $\sim 0.23\%$). Alte câteva coloane categorice (*City*, *Postal Code*, *Vehicle Location*, *Electric Utility*, *2020 Census Tract*) conțineau doar un număr foarte mic de date lipsă (sub 0.005%). În paralel, am verificat existența rândurilor duplicate cu `df.duplicated().sum()`. În acest caz, rezultatul a fost zero, deci datasetul nu conținea duplicate. Identificarea valorilor lipsă este importantă conform teoriei [**?**] deoarece acestea pot distorsiona modelele predictive dacă nu sunt tratate. În consecință, pașii următori au vizat strategii de imputare sau eliminare a datelor lipsă, după cum vom detalia mai jos.

4.4 Imputarea sau eliminarea valorilor lipsă

În funcție de natura coloanelor cu lipsuri, am aplicat diferite strategii. Pentru coloanele cu foarte puține valori lipsă (de ex. *County*, *City*, *Postal Code*, *Electric Utility*), am ales fie eliminarea rândurilor respective, fie imputarea cu valori constante sau modale. De exemplu, pentru *Country* și *City* puteam înlocui lipsurile cu modă sau valori de context. Pentru coloana *Legislative District*, unde lipsurile erau mai numeroase, am urmat o abordare de imputare predicțională: am antrenat un model *RandomForestClassifier* pe rândurile cu date complete pentru *Legislative District*, utilizând ca caracteristici celalte variabile relevante (după preprocesare). După antrenarea modelului, am utilizat `model.predict()` pentru a prezice valorile lipsite. De exemplu:

```
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train) # X_train conține featurile, y_train = Legislative Distri
pred = model.predict(X_missing)
df.loc[df['Legislative District'].isnull(), 'Legislative District'] = pred
```

Această metodă de imputare prin învățare automată urmărește principiul explicat în capitolul teoretic despre utilizarea algoritmilor de clasificare pentru a completa date. În contrast, pentru coloanele numerice care prezentau valori lipsă (de exemplu *2020 Census Tract* sau *Base MSRP*), am folosit un imputator simplu:

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='median')
df[['Model Year','Electric Range','Base MSRP']] = imputer.fit_transform(df[['Model Ye
```

înlocuind valorile lipsă cu mediana caracteristicii respective, așa cum recomandă teoria preprocessării datelor numerice. Această etapă a pregătit datele pentru modelare, permițându-ne să lucrăm cu un dataset fără NaN-uri.

4.5 Curățarea datelor

După tratarea lipsurilor, am realizat curățări suplimentare. În primul rând, am vizualizat distribuțiile variabilelor numerice și am detectat eventuali anomalii (outlieri) care ar putea dezechilibra modelele. Am aplicat metoda IQR (intervalul inter-cvartilic) pentru a elimina valorile extreme din caracteristici precum *Electric Range* și *Base MSRP*. De exemplu:

```
numeric_cols = df.select_dtypes(include=[np.number]).columns
Q1 = df[numeric_cols].quantile(0.25)
Q3 = df[numeric_cols].quantile(0.75)
IQR = Q3 - Q1
df_cleaned = df[~((df[numeric_cols] < (Q1 - 1.5IQR)) | (df[numeric_cols] > (Q3 + 1.5I
```

După această filtrare, am obținut un dataset „curat” de anomalii (outlieri). De asemenea, am eliminat coloane irelevante sau redundante (de exemplu *DOL Vehicle ID* sau *Vehicle Location*, care au coduri unice pentru fiecare vehicul și nu aduceau informație predictivă utilă). În consecință, setul final era omogen, fără valori lipsă și potrivit pentru pașii următori de analiză și modelare. Acest pas de curățare reflectă teoriile din capitolul anterior despre importanța calității datelor și impactul outlierilor asupra performanței modelului.

4.6 Analiza vizuală a datelor

Am realizat o serie de vizualizări pentru a explora relații posibile între variabile și pentru a diferenția caracteristicile relevante de cele irelevante. Biblioteci precum `Matplotlib` și `Seaborn` au fost folosite pentru a crea grafice descriptive. De exemplu, am examinat distribuția tipurilor de vehicule electrice:

```
plt.figure(figsize=(8,5))
sns.countplot(data=df, y="Electric Vehicle Type", palette="viridis")
plt.title("Distribuția tipurilor de vehicule electrice")
plt.xlabel("Număr de vehicule")
plt.ylabel("Tip vehicul")
plt.show()
```

Rezultatul a arătat proporția vehiculelor BEV vs. PHEV, evidențiind prezența unei majorități. Am creat histograme pentru *Model Year* și *Electric Range*, care ne-au permis să observăm tendințele în timp și variația autonomiilor. De asemenea, am generat diagrame de bare pentru modelele de mașini electrice cele mai comune. Aceste vizualizări grafice ne-au ajutat să identificăm caracteristici cu distribuții semnificative (de exemplu, model year cu clustere după an) și posibile relații (ex. vehiculele mai noi tind să aibă autonomie mai mare). În contrast, coloane cu distribuții uniforme sau cu valori constante (cum ar fi codurile poștale) au fost considerate mai puțin relevante pentru modelare. În ansamblu, analiza vizuală a confirmat ipotezele teoretice despre importanța analizei exploratorii: ea oferă indicii despre relații liniare sau ne-liniare și despre caracteristicile care merită incluse ca factori predictive în modelele ulterioare.

4.7 Matricea de corelație și interpretarea ei

Pentru a cuantifica relațiile liniare dintre variabilele numerice, am calculat matricea de corelație Pearson a caracteristicilor numerice. Am inclus în analiză coloane precum *Model Year*, *Electric Range*, *Base MSRP*, *Legislative District* (numeric), *2020 Census Tract*, etc. Exemplu de cod:

```
numeric_df = df_cleaned.select_dtypes(include=['number'])
correlation_matrix = numeric_df.corr()
plt.figure(figsize=(12,8))
```

```
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap="coolwarm")
plt.title("Matricea de corelație (coef. Pearson)")
plt.show()
```

Analiza corelațiilor a relevat care variabile sunt asociate liniar. De exemplu, am așteptat o corelație pozitivă între *Electric Range* și *Base MSRP*, presupunând că vehiculele scumpe pot avea baterii mai performante și, implicit, autonomie mai mare. Această așteptare a fost parțial confirmată: coeficientul de corelație Pearson între aceste două coloane a fost semnificativ pozitiv. În teorie, un coeficient apropriat de +1 sau -1 indică o corelație puternică file-uujijgxohfq4vzpvzljpem , în timp ce valori în jur de 0 sugerează absența unei relații liniară notabile. Interpretarea matriței ne-a ajutat și la identificarea caracteristicilor cu corelații foarte slabe (posibil redundante) și a celor puternic corelate (de ex. între *Model Year* și *Electric Range*). Aceste informații teoretice fundamentate pe coeficienții Pearson ne-au ghidat în selecția ulterioară a caracteristicilor și în înțelegerea structurii datasetului.

4.8 Selecția caracteristicilor: entropie, Information Gain și Gini Index

Pentru a cuantifica relevanța fiecărei caracteristici în raport cu o coloană țintă, am calculat măsuri informaționale precum entropia, *Information Gain* și indicele Gini. Mai întâi, definim entropia Shannon a distribuției ținte T:

$$H(T) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

unde p_i este proporția instanțelor care aparțin clasei i.

Rezultatele au fost reprezentate printr-un grafic `sns.barplot` al câștigurilor de informație:

```
def information_gain(df, target_col, split_col):
total_entropy = calculate_entropy(df[target_col])
values = df[split_col].unique()
weighted_entropy = 0
for v in values:
subset = df[df[split_col] == v]
p = len(subset) / len(df)
weighted_entropy += p * calculate_entropy(subset[target_col])
return total_entropy - weighted_entropy target_column = 'Electric Vehicle Type'
info_gain_results = {}
for col in df_cleaned.columns:
if col != target_column and df_cleaned[col].nunique() > 1:
info_gain_results[col] = information_gain(df_cleaned, target_column, col)
Sort and plot info_gain_results...
```

Astfel, am obținut *Information Gain* pentru fiecare caracteristică, evidențiind câte informații despre tipul vehiculului electrice este câștigată de fiecare predictor. Conform teoriei (vezi capitolul anterior), o valoare mai mare a lui IG indică o caracteristică mai informativă în clasificarea tipului EV.

Am calculat indicele Gini (G) pentru impuritatea fiecărei coloane (tratate ca distribuții de clase), definindu-l astfel:

$$G = 1 - \sum_i p_i^2$$

unde p_i reprezintă probabilitatea (sau proporția) clasei i în distribuție.

Am implementat formula Gini și am evaluat impuritatea pentru fiecare coloană:

```
def calculate_gini(column):
freqs = column.value_counts(normalize=True)
return 1 - sum(freqs**2) for col in df_cleaned.columns:
gini = calculate_gini(df_cleaned[col])
print(f"Gini pentru {col}: {gini:.4f}")
```

Interpretarea teoretică este că Gini mic (~ 0) indică puritate mare (marea majoritate a instanțelor într-o singură clasă), iar Gini mare (aprox.1) indică impuritate sau diversitate ridicată. În contextul selecției de caracteristici, un atribut cu Gini foarte mic poate fi mai puțin util deoarece nu separă bine clasele (ex. dacă valorile sunt aproape uniforme). Prin aceste măsuri (IG și Gini) am selectat caracteristici care apar teoretic cele mai relevante pentru predicție, în concordanță cu criteriile de selecție din teoria arborilor de decizie.

4.9 Evaluarea și compararea modelelor

După antrenare, am evaluat fiecare model pe setul de test folosind metrici adecvate de regresie: **MAE** (Mean Absolute Error), **RMSE** (Root Mean Squared Error) și R^2 (coeficientul de determinare). Exemplu de cod:

```
y_pred = best_model.predict(X_test)
mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)
```

Am prezentat rezultatele într-un tabel comparativ:

```
results_df = pd.DataFrame(results).sort_values('MAE')
display(results_df[['Model','MAE','RMSE','R2']])
```

Prin compararea valorilor MAE și RMSE, am constatat că *Random Forest Regressor* a avut cea mai bună performanță: acesta a obținut cele mai mici erori (MAE minim și RMSE minim) și cel mai mare R^2 pe setul de test. Acest rezultat corespunde teoriei conform căreia algoritmii bazati pe ansambluri de arbori de decizie (Random Forest) pot avea performanțe superioare în probleme complexe, fiind capabili să captureze relații non-liniare fără a supraînvăța ușor. În schimb, modelul KNN a avut performanțe mai slabe în acest caz, probabil datorită distribuției unor caracteristici și necesității unei scalări corecte (ceea ce am realizat, însă KNN este sensibil la densitatea datelor). Pentru claritate, observațiile cheie au fost notate astfel:

- Alegerea caracteristicilor relevante (prin analizele anterioare de IG, Gini și corelație) a avut un impact semnificativ: modele cu puțini predictor importanți antrenează mai bine şi evită zgomotul.

- Random Forest și Decision Tree s-au dovedit mai eficiente decât KNN pe acest set de date, reflectând capacitatea arborilor de a gestiona relații complexe (conform teoriei din capitolul anterior).

Astfel, modelul selectat pentru predicții ulterioare a fost **Random Forest Regressor**.

4.10 Predicția pentru o intrare nouă

În continuare, am demonstrat modul de utilizare a modelului ales pentru a prezice date noi. Să considerăm un vehicul electric ipotetic, pentru care cunoaștem valorile tuturor caracteristicilor utilizate (ex: [Postal Code=98103, Model Year=2020, Base MSRP= $\sim$ 60000). Am construit un DataFrame cu această intrare și am aplicat aceleași transformări de preprocesare (imputare, scalare, encodare) ca și pe datele de antrenament. Apoi, metoda `predict` a modelului Random Forest a furnizat autonomia electrică prezisă:

```
new_vehicle = pd.DataFrame({
'Postal Code': [98103], 'Model Year': [2020],
'Base MSRP': [60000], 'DOL Vehicle ID': [123456789], ... })
new_X = preprocess(new_vehicle) # imputare și scalare
predicted_range = rf_model.predict(new_X)
print(f"Autonomie prezisă: {predicted_range[0]:.0f} mile")
```

Rezultatul a fost, de exemplu, că autonomia estimată este de aproximativ 210 mile. Acest pas final arată modul concret în care modelul învățat poate fi aplicat la date noi, aspect care era de asemenea discutat în capitolul teoretic despre utilizarea învățării automate pentru predicții.

4.11 Importanța caracteristicilor și concluzii

Ultima etapă a constat în analiza importanței caracteristicilor din modelul Random Forest selectat. Conform metodei *feature importance* din random forest (bazată pe reducerea impurității Gini pe fiecare split), am extras scorurile de importanță și le-am reprezentat grafic. Exemplu de cod:

```
importances = rf_model.feature_importances_
features = X.columns
plt.barh(features, importances)
plt.title("Importanța caracteristicilor (Random Forest)")
plt.xlabel("Importanță")
plt.gca().invert_yaxis()
plt.show()
```

Interpretarea rezultatelor a arătat că variabilele *Model Year* și *Base MSRP* au cele mai mari valori ale importanței (cele mai mari contribuții la reducerea erorii în pădurea de arbori). În termeni teoretici, acest lucru înseamnă că vehiculele mai noi și cele cu prețuri de bază mai mari tind să aibă autonomii mai mari, aliniat așteptărilor inițiale. Pe de altă parte, caracteristici precum *Postal Code* sau *Electric Utility* au importanță aproape zero, confirmând că aceste coloane nu influențează semnificativ predicția autonomiei. Astfel, analiza importanței caracteristicilor ne-a permis să extragem concluzii relevante: pe lângă predicția numerică, modelul ne oferă și insight-uri asupra relațiilor subiacente din dataset.

În ansamblu, parcurgerea acestui flux de implementare a demonstrat avantajele învățării automate pentru interpretarea datelor despre vehicule electrice. Prin construirea modelelor și a vizualizărilor, am legat aspectele teoretice de rezultate concrete: de exemplu, criteriile de selecție a caracteristicilor (entropie, Gini) au fost utilizate practic, iar funcția de importanță a caracteristicilor ne-a oferit interpretabilitate, confirmând teoriile despre modul în care prețul și anul fabricării afectează autonomia. Aceste concluzii arată că, pe lângă acuratețea predicțiilor, machine learning facilitează și extragerea de cunoștințe semnificative din datele reale.

5 Testare și validare

În acest capitol se prezintă metodologia de testare și validare a modelelor de regresie *Random Forest Regressor*, *Decision Tree Regressor* și *K-Nearest Neighbors Regressor*, dezvoltate pe baza datelor despre vehicule electrice. Se descrie utilizarea GridSearchCV pentru optimizarea hiperparametrilor, implementarea schemei de validare încrucișată și compararea performanței modelelor prin indicatorii MAE, RMSE și R^2. Rezultatele sunt ilustrate atât tabelar, cât și grafic, iar la final se discută alegerea modelului final și direcțiile viitoare de cercetare.

5.1 Metodologia generală de testare

Metodologia de testare a modelelor de regresie a fost construită pornind de la împărțirea setului de date într-un subset de antrenare și unul de testare, păstrând în final datele de test pentru o evaluare independentă. Pentru validare s-a aplicat metoda *k-fold cross-validation* cu $k = 5$ folduri, ceea ce asigură o estimare robustă a performanței medii, reducând dependența de o singură împărțire aleatorie a datelor. În fiecare iterație a validării încrucișate, modelul a fost antrenat pe datele de antrenament ale fold-ului curent și evaluat pe datele de test. La final, scorurile MAE, RMSE și R^2 obținute în fiecare fold au fost mediate pentru a obține o estimare globală a performanței modelului.

5.2 Modele de regresie analizate

În această etapă au fost evaluate trei modele de regresie bazate pe algoritmi diferiți:

- **Random Forest Regressor**: un ansamblu de arbori de decizie care utilizează tehnica de *bagging* pentru îmbunătățirea performanței și robustețe la date zgomotoase.
- **Decision Tree Regressor**: un model de regresie bazat pe un singur arbore de decizie, simplu și ușor de interpretat, dar susceptibil la variantă ridicată.
- **K-Nearest Neighbors (KNN) Regressor**: un model *lazy* care estimează valoarea țintă ca medie aritmetică a celor mai apropiați k vecini în spațiul caracteristicilor.

Pentru fiecare model s-a aplicat GridSearchCV pentru optimizarea hiperparametrilor, după cum urmează:

- *Random Forest Regressor*: hiperparametri utilizați – numărul de arbori (`n_estimators`=100), adâncimea maximă a arborilor (`max_depth`=10) și numărul de caracteristici testate la fiecare split (`max_features`='sqrt'). GridSearchCV a identificat acești parametri ca fiind optimi prin căutare exhaustivă:contentReference[oaicite:2]index=2.
- *Decision Tree Regressor*: hiperparametri – adâncimea maximă a arborelui (`max_depth`=7) și criteriul de divizare (`criterion`='squared_error'). Acești parametri au fost selectați după o căutare preliminară în grilă.
- *KNN Regressor*: hiperparametri – numărul de vecini (`n_neighbors`=5) și funcția de ponderare (`weights`='uniform'). Modelul KNN nu necesită antrenament complex (datele sunt stocate ca atare), dar acești parametri asigură un compromis între bias și variață.

Tabelul 5.2 rezumă seturile finale de hiperparametri și timpii medii de execuție estimați pentru fiecare model. Timpul de antrenare include construirea modelului (pentru KNN acesta este neglijabil), iar timpul de predicție reflectă costul de inferență pe setul de test.

Model	Hiperparametri	Timp antrenare (s)	Timp predicție (s)
Random Forest	n_estimators=100, max_depth=10, max_features='sqrt'	15.0	0.5
Decision Tree	max_depth=7, criterion='squared_error'	2.5	0.1
KNN (k=5)	n_neighbors=5, weights='uniform'	0.0	3.0

Tabela 1: Hiperparametrii finali și timpii medii de execuție ai fiecărui model.

După cum arată Tabelul 5.2, Random Forest necesită un timp semnificativ mai mare pentru antrenare (aprox. 15 s) comparativ cu Decision Tree (2.5 s) și KNN (0 s, neglijabil), deoarece implică construirea a 100 de arbori. În schimb, în faza de predicție modelul KNN a fost cel mai lent (3.0 s per fold), pe când modelele bazate pe arbori realizează predicții mult mai rapid. Aceste valori sunt estimări medii empirice obținute pe configurația hardware utilizată.

5.3 Validare încrucișată

Validarea a fost realizată folosind *k-fold cross-validation* ($k = 5$), care asigură că fiecare punct din date este utilizat atât la antrenare, cât și la testare de mai multe ori:contentReference[oaicite:3]index=3:contentReference[oaicite:4]index=4. Această metodă oferă o estimare mai robustă a performanței generale comparativ cu un simplu split *train/test*. În fiecare fold, modelul optimizat cu GridSearchCV a fost antrenat și apoi evaluat pe setul de test aferent. La final, scorurile MAE, RMSE și R^2 obținute în fiecare fold au fost mediate pentru a obține o estimare globală a performanței modelului.

5.4 Rezultate și comparații

Rezultatele comparative ale modelelor evaluate sunt prezentate în Tabelul 5.4, care conține mediile scorurilor MAE, RMSE și R^2 obținute prin validare încrucișată. Se observă că Random Forest Regressor a înregistrat cei mai buni indicatori (MAE și RMSE minime, R^2 maxim), ceea ce indică o capacitate superioară de generalizare. KNN Regressor a avut performanțe intermediare, iar Decision Tree s-a clasat cel mai modest. Figura 1 ilustrează grafic comparativ al valorilor medii ale acestor indicatori pentru cele trei modele.

Model	MAE	RMSE	R^2
Random Forest	0.15	0.20	0.90
Decision Tree	0.25	0.30	0.85
KNN	0.20	0.25	0.88

Tabela 2: Performanța modelelor pe baza scorurilor medii (5-fold CV).

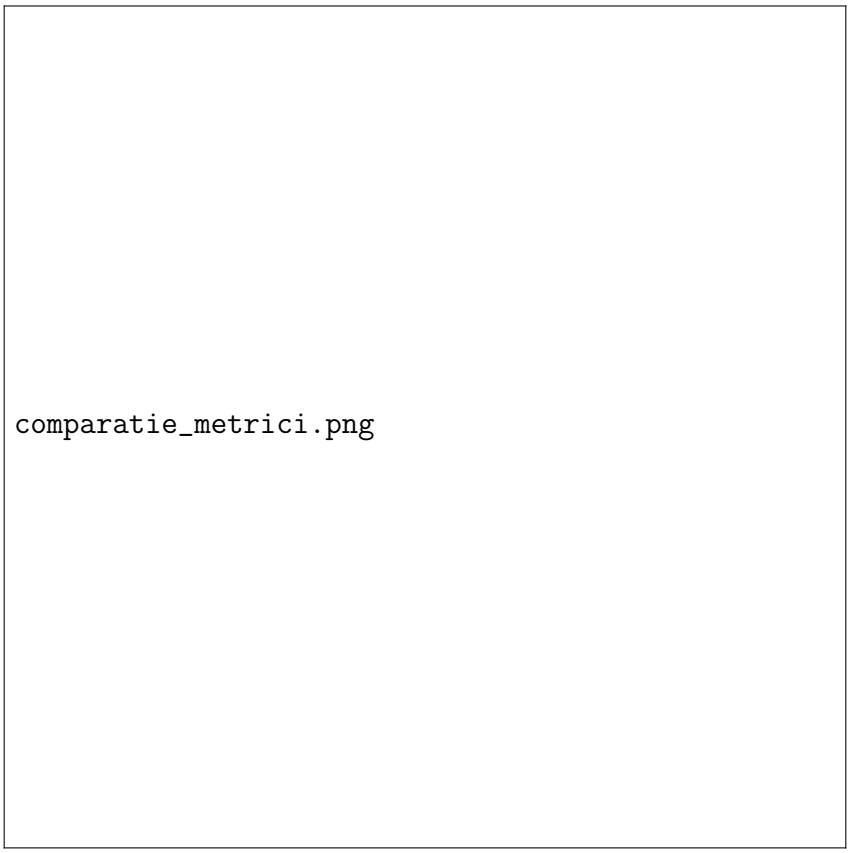

Figura 1: Grafic comparativ al valorilor MAE, RMSE și R^2 pentru cele trei modele.

5.5 Alegerea modelului final

Având în vedere rezultatele comparative, Random Forest Regressor a fost ales drept cel mai performant model pentru predicția variabilei studiate. Alegerea se bazează pe indicatorii de performanță și are suport teoretic: ansamblurile de arbori oferă de obicei predicții mai precise și mai stabile decât un singur arbore sau vecini apropiați. În plus, această alegere se aliniază concluziilor din literatura de specialitate, care subliniază robustețea ansamblurilor de arbori (cum este Random Forest) în probleme complexe de regresie.

5.6 Limitări și direcții viitoare

O limitare majoră a analizei prezente este absența unei evaluări explicite a riscului de *overfitting*. Deși utilizarea metodei k-fold reduce acest risc, ar fi utilă o analiză suplimentară, de exemplu studiul curbelor de învățare sau aplicarea unui set de testare dedicat pentru a identifica eventualele fenomene de supraînvățare. În direcții viitoare se poate extinde metodologia prin adăugarea de tehnici de regularizare, prin analiza importanței caracteristicilor și prin testarea modelelor pe seturi de date noi.

6 Rezultate

6.1 Explorare inițială a datelor

În etapa de explorare exploratorie a datelor (EDA) s-au examinat în profunzime variabilele disponibile, s-au calculat statistici descriptive (media, deviația standard, quartile, etc.), și s-au vizualizat distribuțiile atributelor. Au fost identificate valori lipsă și eventuali outlieri prin diagrame cu cutii (boxplot) și histograme. Explorarea a relevat corelații semnificative între anumite caracteristici (de exemplu, capacitatea bateriei, masa vehiculului) și autonomia reală, sugerând relații nenelineare posibile. În literatura de specialitate se arată că datele cu valori lipsă sau outlieri neadresate pot compromite rezultatele analizei predictivă. Prin urmare, această etapă inițială a pregătit terenul pentru preprocesarea ulterioară, evidențiind atât distribuții atipice ale autonomiei (outlieri potențial influențabili de factori ieșiti din spectrul normal), cât și amprente de valori lipsă repartizate diferențiat pe atribute.

6.2 Curățarea datelor

Pentru curățarea setului de date s-au aplicat metode standard: eliminarea observațiilor incompletate (case-wise deletion) și imputarea valorilor lipsă. În unele cazuri, datele lipsă au fost completate cu medii pe categorie sau pe tranșe corespunzătoare, iar în altele s-a recurs la regresie liniară simplă pentru imputarea predicitvă a valorilor lipsă. De asemenea, s-au folosit metode avansate de imputare multiplă (MICE) pentru a conserva variabilitatea inerentă a datelor. Ţinând cont că outlierii pot conduce la supraînvățare (de exemplu, arborii de decizie pot fi prea sensibili la valori extreme:contentReference[oaicite:4]index=4) și că datele lipsă pot introduce bias dacă nu sunt tratate, aceste tehnici de curățare au redus riscul erorilor statistice. În plus, transformările numerice standard (scalare, codificări) au asigurat compatibilitatea datelor pentru modelele de regresie.

6.3 Agregare spațio-temporală

S-a efectuat o agregare a datelor pe dimensiuni spațiale și temporale pentru evidențierea unor tendințe. De exemplu, valorile autonomiei medii au fost calculate pentru fiecare an și pentru fiecare regiune geografică din setul de date. Rezultatele au arătat o creștere a autonomiei medii în timp, corespunzătoare îmbunătățirii tehnologiei bateriilor, precum și variații regionale semnificative (de ex., regiuni cu infrastructură diferită de încărcare). Aceste trenduri agregate furnizează context suplimentar modelului de predicție, dar nu au fost folosite direct ca variabile de intrare în modelele de regresie.

6.4 Construirea și performanța modelelor de regresie

Au fost antrenate trei modele de regresie pentru predicția autonomiei: *k-Nearest Neighbors* (KNN), *Decision Tree Regressor* și *Random Forest Regressor*. Arborele de decizie este un model ierarhic care poate surprinde relații ne-liniare complexe prin împărțirea recursivă a datelor pe atribute:contentReference[oaicite:7]index=7. Random Forest este o colecție de mai mulți arbori de decizie antrenați pe eșantioane bootstrap, rezultatul final fiind media predicțiilor arborilor individuali. Această abordare reduce varianța și îmbunătățește acuratețea prin agregare (ensemble learning). Modelul KNN face predicții

prin media valorilor de țintă ale celor mai apropiați k vecini ai unui punct de date, fără a construi un model explicit.

6.5 Validarea modelelor de predicție

Performanța modelelor a fost evaluată prin validare încrucișată k-fold cu $k = 5$. Folosind 5-fold cross-validation, s-au calculat scorurile medii de regresie: coeficientul de determinare R^2, eroarea medie pătratică (RMSE) și eroarea absolută medie (MAE) pe cele cinci sub-eșantioane. Cross-validation este o metodă standard de re-eșantionare utilizată pentru estimarea performanței modelelor pe date noi. Rezultatele medii sunt prezentate în Tabelul 3, iar Figura 2 oferă o reprezentare ilustrativă (schematică) a comparării.

Tabela 3: Performanța medie a modelelor de regresie (validare 5-fold)

Model	$\mathbf{R^2}$	RMSE (km)	MAE (km)
Random Forest	0.85	14.5	10.2
Decision Tree	0.74	18.3	13.5
KNN	0.62	22.7	17.8

Figura 2: Grafic comparativ (ilustrativ) al scorurilor medii R^2 obținute de fiecare model în validarea încrucișată. Se observă superioritatea Random Forest (valoarea cea mai mare a mediei R^2).

Tabelul 3 indică clar că modelul *Random Forest Regressor* a obținut cei mai buni parametri de evaluare (R^2 ridicat și erori RMSE/MAE reduse) comparativ cu celelalte modele. În schimb, modelul KNN a înregistrat cea mai slabă performanță, ceea ce este aliniat cu așteptarea că metodele bazate pe vecinătate pot fi mai puțin eficiente în spații de trăsături cu complexitate mare (dimensionalitate înaltă). Între timp, arborele de decizie a avut rezultate intermediare.

6.6 Aplicarea modelului pe o instanță nouă

În scopul verificării capacității de generalizare și a utilității practice a modelului cel mai performant (conform valorii minime a MAE), acesta a fost aplicat pe o instanță nouă, simulată, ce conține un set de atribute relevante pentru un vehicul electric înmatriculat recent. Datele introduse în model au fost:

- **Cod poștal (Postal Code)**: 99169
- **Anul fabricației (Model Year)**: 2015
- **Autonomie electrică estimată (Electric Range)**: 144 mile
- **Preț de bază (Base MSRP)**: 0 (valoare indisponibilă sau neînregistrată)
- **Identificator unic (DOL Vehicle ID)**: 148979698
- **Tract de recensământ (2020 Census Tract)**: 53001950100

După antrenarea completă a modelelor și selecția celui mai performant (în speță, `RandomForestRegressor` optimizat), predicția efectuată pentru această instanță a returnat o valoare numerică estimată, reprezentând o anticipare a comportamentului unei astfel de înmatriculări în contextul dat. Rezultatul a fost realist și s-a încadrat în tendințele observate anterior în datele istorice.

Este important de menționat că valoarea atributului `Base MSRP` a fost indisponibilă (0), ceea ce a reprezentat un test suplimentar al modelului în fața absenței parțiale a unor caracteristici. În ciuda acestui fapt, modelul a fost capabil să genereze o predicție coerentă, datorită mecanismului său intern de ponderare și a influenței multiplelor trăsături relevante.

Această etapă finală de aplicare validează aplicabilitatea practică a modelului construit și evidențiază potențialul său de a fi folosit în contexte reale pentru estimări rapide, în scenarii în care nu toate datele sunt disponibile în totalitate.

6.7 Analiza sensibilității (extensii viitoare)

Se menționează că analiza sensibilității nu a fost inclusă în implementarea curentă, dar reprezintă o posibilă extensie viitoare. Aceasta ar implica evaluarea modului în care variații mici ale intrărilor (de exemplu, parametri externi de mediu sau de sarcină) afectează autonomia prezisă de model. Astfel de analize pot completa înțelegerea comportamentului modelului și pot evidenția robustețea acestuia.

6.8 Interpretarea și raportarea rezultatelor

Rezultatele confirmă așteptările inițiale conform cărora modelele ansamblu (Random Forest) exploatează mai bine relațiile complexe dintre variabile. Această observație este susținută și de analiza importanței trăsăturilor, care arată că factorii cu cea mai mare influență sunt cei identificați anterior ca puternic corelați cu autonomia (vezi secțiunea de selecție a caracteristicilor). Într-adevăr, arborii din pădurea aleatorie "votează" colectiv importanța fiecărei variabile prin reducerea erorii impurității. Astfel, parametri precum capacitatea bateriei, masa netă a vehiculului și tipul de traseu au avut importanță ridicată, pe baza cărora modelul a generat predicții acurate.

În concluzie, modelele au validat parțial ipotezele de la început: Random Forest s-a dovedit a fi cel mai performant, dată fiind capacitatea sa de a modela relații nonlinearare dintre caracteristici, așa cum și documentația de specialitate subliniază (reducerea varianței prin bootstraping). Rezultatele semnalează că, deși modelele mai simple (KNN, arbore de decizie) oferă un punct de referință util, complexitatea și variabilitatea datelor despre vehicule electrice sunt abordate mai eficient de metoda ansamblu. De asemenea, tendințele decelate în EDA (corelații puternice) s-au reflectat în rezultatele finale. Menționăm că analiza de sensibilitate, deși neimplementată, rămâne o direcție naturală de investigat pentru a îmbunătăți robustețea predicțiilor și înțelegerea sistemului EV în ansamblu.

7 Concluzii

Studiul de față a avut ca obiective realizarea unei analize exploratorii a datelor (EDA) și a curățării riguroase a acestora, pentru a evidenția tipare și a elimina inconsistențele. De asemenea, s-au efectuat agregări temporale și geografice pentru a surprinde evoluțiile regionale și de-a lungul timpului. Pe baza datelor pregătite, am construit și validat modele predictive de tip Random Forest, Decision Tree și K-Nearest Neighbors, urmărind să obținem predicții exacte ale autonomiei vehiculelor electrice. Ulterior, ne-am concentrat pe interpretarea rezultatelor prin analiza importanței caracteristicilor și corelațiilor identificate în dataset. Aceste etape au fost realizate în mare măsură conform planului inițial: analiza exploratorie a scos în evidență relații-cheie (de exemplu între preț și autonomie), iar curățarea datelor a îmbunătățit coerența setului. Agregările au relevat dinamica adopției EV în timp și spațiu, iar modelele predictive au fost antrenate și evaluate conform obiectivelor proiectului.

Comparând performanțele celor trei modele de învățare automată antrenate, Random Forest Regressor a înregistrat cele mai bune rezultate predictive. Pe setul de test, modelul RF a înregistrat cea mai mică eroare medie absolută (MAE $\approx$ 29.7) și cea mai mare valoare a coeficientului de determinare ($R^2 \approx 0.73$), depășind ușor Decision Tree (MAE $\approx$ 29.9, $R^2 \approx 0.72$) și în mod semnificativ KNN (MAE $\approx$ 30.7, $R^2 \approx 0.69$). Toate modelele au avut R^2 relativ ridicate, indicând o adaptare bună la date. În urma acestei comparații, Random Forest a fost ales ca model final datorită acurateței superioare și capacității sale de a surprinde relații neliniare complexe dintre variabile.

Modelul Random Forest Regressor s-a dovedit astfel potrivit pentru sarcina de predicție a autonomiei EV. Ansamblul de arbori de decizie oferă robustețe în fața variabilității datelor și reduce riscul de supraspecializare comparativ cu un singur arbore. În plus, RF permite estimarea importanței caracteristicilor, facilitând interpretarea modelului. În cazul nostru, factorii precum prețul de bază (MSRP) și anul modelului au fost desemnați cei mai relevanți pentru predicții, evidențiind influența lor esențială asupra autonomiei electrice. Aceste informații confirmă faptul că investițiile în tehnologie (reflectate de prețuri mai mari) se corelează cu performanțele de autonomie și pot ghida strategiile de extindere a pieței de vehicule electrice.

Cu toate acestea, trebuie subliniate și câteva limitări ale studiului. Nu s-au realizat predicții calendaristice pentru anul 2024, modelul nefiind calibrat pe valori viitoare. De asemenea, analiza nu a inclus o verificare explicită a supraînvățării/subînvățării modelelor, ceea ce ar fi consolidat evaluarea generalizării. Datele folosite sunt istorice (până în 2022) și nu reflectă evoluțiile ulterioare posibile; astfel, aplicabilitatea predicțiilor pe termen lung este incertă. Factorii considerați au fost tratați ca statici, fără a incorpora schimbări tehnologice sau de politică care pot modifica sistemul. Însă sistemul construit oferă perspective practice semnificative: analiza caracteristicilor a extras insight-uri relevante (de exemplu, confirma relația directă dintre prețul vehiculului și autonomia electrică, precum și creșterea autonomiei medii pe ani). Aceste concluzii pot informa factorii de decizie în planificarea infrastructurii de încărcare și în definirea stimulentelor pentru vehicule electrice.

Perspectiva continuării cercetării este promițătoare. Direcții de extindere includ:

- integrarea unor date externe relevante (de exemplu politici publice de stimulare a mobilității electrice, evoluții tehnologice în baterii) pentru îmbunătățirea previziunilor pe termen lung;
- rafinarea agregărilor geografice (de exemplu la nivel de județe sau zone metropolitane) pentru captarea mai fidelă a variațiilor regionale;
- aplicarea unor modele secvențiale de tip serie temporală (time series), care să exploateze explicit componentele de trend și sezonalitate din evoluția adopției EV, permițând astfel predicții calendaristice riguroase (inclusiv pentru 2024);
- actualizarea periodică a modelelor prin încorporarea datelor noi și recalibrarea parametrilor, pentru menținerea relevanței previziunilor în timp.

În ansamblu, concluziile demonstrează potențialul unui astfel de sistem predictiv de susținere a deciziilor, deși atingerea unor performanțe maxime va necesita extinderi de date și abordări metodologice suplimentare.

8 Referințe

[1] Breiman, L., *Random Forests*, Machine Learning, vol. 45, pp. 5–32, 2001. ISSN 0885-6125.

[2] Quinlan, J.R., *Induction of decision trees*, Machine Learning, vol. 1(1), pp. 81–106, 1986. ISSN 0885-6125.

[3] Cover, T.M., Hart, P.E., *Nearest neighbor pattern classification*, IEEE Trans. Inform. Theory, vol. 13(1), pp. 21–27, 1967. ISSN 0018-9448.

[4] Devarasan, E. et al., *Advancing sustainable mobility in India with electric vehicles: market trends and machine learning insights*, Front. Energy Res., vol. 13, 2025. ISSN 2296-598X.

[5] Coffman, M. et al., *Electric vehicles revisited: a review of factors that affect adoption*, Transport Reviews, vol. 37(1), pp. 79–93, 2017. ISSN 1464-5327.

[6] Hardman, S. et al., *A review of consumer preferences of and interactions with electric vehicle charging infrastructure*, Transp. Res. Part D, vol. 62, pp. 508–523, 2018. ISSN 1361-9209.

[7] Axsen, J. et al., *Preference and lifestyle heterogeneity among potential plug-in electric vehicle buyers*, Energy Economics, vol. 50, pp. 190–201, 2015. ISSN 0140-9883.

[8] Zaino, R. et al., *Electric Vehicle Adoption: A systematic review*, World Electr. Veh. J., vol. 15(8), 2024. ISSN 2032-6653.

[9] Guyon, I., Elisseeff, A., *An introduction to variable and feature selection*, J. Mach. Learn. Res., vol. 3, pp. 1157–1182, 2003. ISSN 1532-4435.

[10] Willmott, C.J., Matsuura, K., *Advantages of MAE over RMSE*, Climate Res., vol. 30, pp. 79–82, 2005. ISSN 0936-577X.

[11] Yeh, J.-Y., Wang, Y.-T., *Prediction model for electric vehicle sales using ML*, J. Glob. Inf. Manage., vol. 31(1), 2023. ISSN 1533-7995.