Raport tehnic

1.**Curatarea datelor**

Dupa analiza datelor de test am ales pastrarea urmatoarelor coloane ["data, "Stare", "Oferit de", "Are VIN (Serie sasiu)", "Marca", "Model", "Versiune", "Anul fabricației", "Km", "Combustibil", "Putere", "Cutie de viteze", "Consum Extraurban", "Transmisie", "Consum Urban", "Tip Caroserie", "Emisii CO2", "Numar de portiere", "Numar locuri", "Se emite factura", "Eligibil pentru finantare", "Garantie dealer (inclusa in pret)", "Primul proprietar (de nou)", "Fara accident in istoric", "Carte de service","Audio si tehnologie", "Confort si echipamente optionale", "Electronice si sisteme de asistenta","Siguranta", "Culoare", "Vehicule electrice"], deoarece prezinta informatii relevante pentru determinarea pretului.

Am prelucrat coloanele dupa cum urmeaza:

- Anul fabricației
  convertit la int si inlocuit datele lipsa cu 0
- Starea
  Mapat dupa cum urmeaza „Nou” -> 1 si „Secoond hand” -> 0
- Are VIN (Serie sasiu)
  Ai marcat 1 dacă valoarea este "Da", altfel 0.
- Putere, Km, Consum Extraurban, Consum Urban, Emisii CO2, Numar de portiere, Numar locuri
  Convertite la int, iar valorile lipsa au valoarea mediei
- Se emite factura, Eligibil pentru finantare
  Convertite la binar: "Da" → 1, altfel 0.
- Garantie dealer (inclusa in pret)
  Eliminare luni si converitre la luni, 0 daca lipseste
- Primul proprietar (de nou), Fara accident in istoric, Carte de service
  1 dacă există o valoare (nu e NaN), altfel 0.
- Siguranta, Electronice si sisteme de asistenta, Vehicule electrice, Audio si tehnologie, Confort si echipamente optionale
  Pentru fiecare element din lista de feature-uri construim o noua coloanal cu valuarea 1 daca se gaseste in lista, 0 altfel
- Oferit de, Marca, Model, Combustibil, Tip Caroserie, Culoare, Cutie de viteze, Versiune, Transmisie
  Am eliminat cuvintele de legătură (stopwords) și caracterele speciale. Apoi le-am tokenizat folosind word_tokenize, iar ulterior le-am convertit în vectori numerici folosind modelul Word2Vec / modelul transformer intfloat/e5-small

Dupa toti acesti pasi coloanele au fost normalizate folosind un scaler. Aceleasi transformari se aplica si pe datele de test.

2**. Modele de regresie**

Am impartit datele in 80% pentru antrenament si 20% test.

In ceea ce priveste modelele de regresie folosite am ales doua abordari un folosind GradientBoostingRegressor, RandomForestRegresor si XGBoost.

3**. Hyperparameters tuning**

Acest pas a fost realizat doar pentru modelul XGBoost deoarece a oferit preziceri mult mai bune fara tooning. Acest pas a fost realizat in doua moduri de mana si folosind grid search asupra a 3 hyperparametri: numar de estimatori, max depth si learning depth. Valorile cele mai bune de predictie fiind obtinute cu valorile de 4000 estimatori, 8 max_depth si un llearning rate de 0.01.

4.**Findings:**

Folosind e5-small pentru vectorizarea denumirilor rezulta in predictii semnificativ mai precise, acest lucru fiind oferit si de faptul ca modelul este preantrenat, pe cand varianta de w2vec incercata a fost antrenata pe toate valorile din coloanele la care a fost folosit.

Rularea codul s-a facut pe colab, acest lucru rezolvand problema rularii pe local cauzate de tensorflow, dar vine la pachet cu probleme de deconectare de la sesiune, deci pierderea intregului progres al rularii.