Autóár előrejelzés lineáris regresszió és véletlen erdő modellek használatával

Gutter-Bacsi Zsombor

December 14, 2025

Abstract

Ez a projekt egy Kaggle adathalmaz felhasználásával készített használt autók árának előrejelzését mutatja be, adatszűréssel és jellemzőkiemeléssel kombinálva lineáris regressziót és véletlen erdő modelleket. A véletlen erdő érte el a legjobb pontosságot, míg a lineáris regresszió érzékeny volt a kiugró értékekre. Az összes kód elérhető a GitHub-on, és a jövőbeli munkák között szerepel egy árazó API üzembe helyezése.

1 Bevezetés

Ehhez a projekthez úgy döntöttem, hogy egy valós és némileg zavaros adathalmazzal dolgozom a Kaggle-ről. Több jelölt átnézése után a *Car Price Prediction Challenge* adathalmazt választottam [1]. Nem veszek részt a kihívásban, de a feladatleírást használtam a munkám irányító elképzelésének: építsünk egy modellt, ami egy használt autó árát előrejelzi attribútumai alapján.

Célom volt elérni egy modellt, amely legalább 80%-os gyakorlati előrejelzési pontosságot ér el. Később megtanultam, hogy a választott metrika függvényében ez a szám lehet értelmes vagy félrevezető. Más szavakkal, egy modell lehet statisztikailag helyes, mégis rosszul viselkedhet valós autók esetében.

2 Módszerek

Munkafolyamatom négy fő szakaszon keresztül fejlődött: adatkezelés, feltáró adatelemzés (EDA), a sztring kódoló fejlesztése, és végül a regressziós modellek. Bár ezeket a lépéseket logikus sorrendben mutatom be, a gyakorlatban mindig visszatértem, amikor modellezési problémára bukkantam. Sok olyan ötlet, amely nem működött, ugyanolyan fontos volt, mint azok, amelyek működtek, mert ők alakították a végső tervezetet.

2.1 Adatkezelés

A projekt a nyers Kaggle CSV betöltésével kezdődött egy egyéni `CarRecord` szerkezetbe, amely később egy `Car` objektumot hozott létre, amely tartalmazott egy `Engine`-t. Szándékom az volt, hogy az összes adatot egy rendezett, objektum-orientált formátumban tartsam. Hosszú távon ez az ötlet még mindig potenciállal bír, mert a beágyazott attribútumok (például motoradatok) természetesen kezelhetők objektumokban.

Azonban amint gyakorlati EDA-t kezdtem, világossá vált, hogy a `pandas DataFrame` sokkal hatékonyabb. A legtöbb ábrázolás, szűrés és statisztikai művelet drámaian könnyebb volt. Ennek eredményeként a DataFrame vált a fő munkareprezentációvá, míg az objektummodell jövőbeli bővítés koncepciójaként maradt.

2.2 Feltáró adatelemzés

Az EDA-t egyszerű statisztikákkal kezdtem az árakról és más numerikus mezőkről. Az ár eloszlása rendkívül széles tartományt mutatott, gyakorlatilag nullától több mint 200 000 USD-ig.

A 1. ábrán látható korrelációs hőtérkép több fontos mintát mutatott. Míg néhány változó csak gyenge közvetlen korrelációt mutatott az árral (például a kilométeróra, körülbelül $-0,01$), mások, mint a gyártási év, mérsékelt pozitív korrelációt mutattak ($\approx 0,40$). Még fontosabb, hogy a *teljes* jellemzőkészlet használata közvetett kapcsolatokat tár fel: a gyártási év korrelál olyan jellemzőkkel, mint a belsőőr bőrburkolat, amelyek maguk is korrelálnak az árral. Ezek a többlépcsős függőségek nem lennének láthatók egy csökkentett hőtérképen, amely csak az árat és néhány numerikus változót tartalmazza. Ez látható az ár és a gyártási év páronkénti korrelációjában, amely önmagában gyengének tűnik, míg a teljes hőtérkép világosabb kapcsolatot mutat, ha figyelembe vesszük más jellemzőkkel való interakciókat.

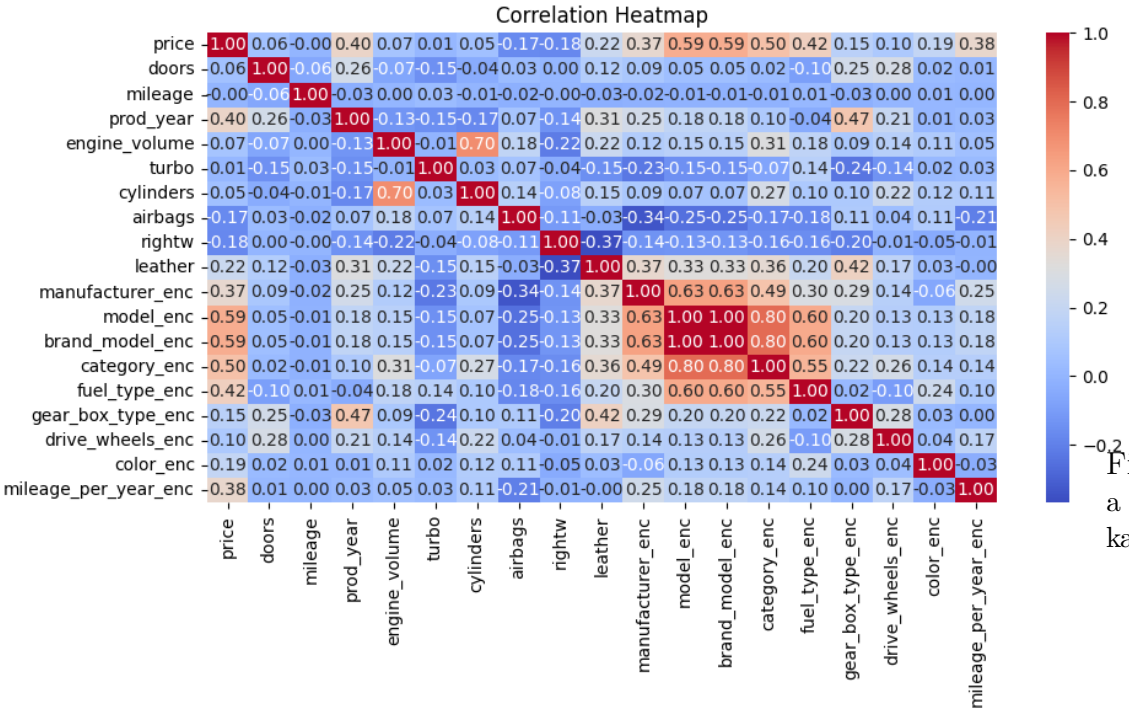

Figure 1: Összes numerikus és kódolt kategorikus jellemző korrelációs hőtérképe.

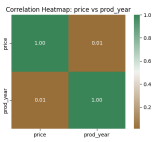

Figure 2: Az ár és a gyártási év páronkénti kapcsolata.

Az ár hisztogram és a kategória eloszlási ábrák szinte minden későbbi szűrési döntést vezéreltek.

A gyártó eloszlása a 4. ábrán erős egyensúlyhiányt mutatott: sok márka csak néhányszor jelent meg. A kódolási folyamat stabilizálása érdekében minden gyártó és modell kategóriát, amelynek kevesebb mint 180 bejegyzése volt, kiszűrtem.

Egy másik kulcsfontosságú felfedezés az árak eloszlásából származott, amit a 3. ábra mutat. Meglepően nagy számú autó volt "piszok olcsó" (gyakran 1000 USD alatt), míg egy kisebb, de jelentős csoport rendkívül drága volt. Ezek a szélsőségek nem úgy viselkednek, mint normális piaci árak, és nagyon nagy előrejelzési hibákat produkáltak. A legalacsonyabb és legmagasabb kiugró értékek eltávolítása a $\ln(\text{ár})$ küszöbértékek alapján (pl. $\ln(p) < 7$ vagy $\ln(p) < 8$ és $\ln(p) > 12$) ezért döntő fontosságú volt mindkét regressziós modell stabilizálásához.

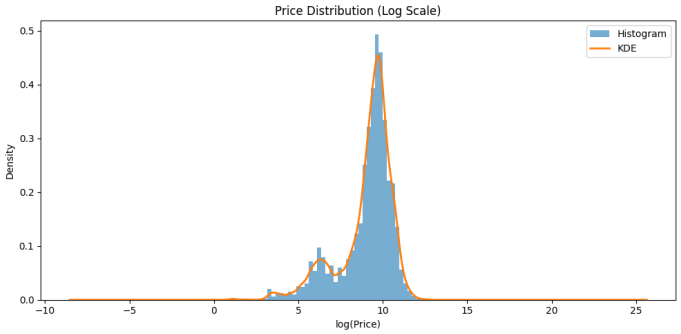

Figure 3: Ár eloszlási hisztogram

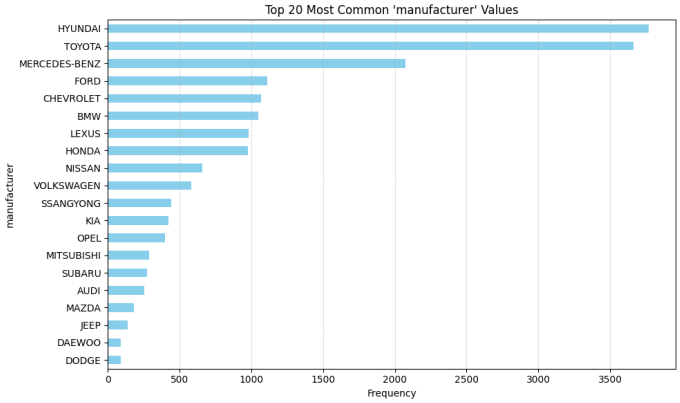

Figure 4: Gyártó eloszlása

2.3 Sztring kódolás és jellemzőkiemelés

A kódoló vált az egész projekt központi összetevőjévé. Mindkét regressziós modell nagymértékben függött ezen jellemzők minőségétől.

(1) Átlagár-alapú kódolás

A kategorikus változók naiv egész számú kódolása (pl. gyártó $= \{0, 1, 2, \dots\}$) értelmetlen volt, mert rangsort jelentene a márkák között. Ehelyett minden kategóriához hozzárendeltem az adott kategóriában lévő autók *átlagárát*. Például a BMW természetesen magasabb kódolt értéket kap, mint egy költségvetési márka, ha a BMW-k általában drágábbak.

Ez nagyon jól működött nagy kategóriáknál, mint a gyártó vagy modell, és meglehetősen jól még kisebb kategóriáknál is, mint az üzemanyag típus vagy a sebességváltó típusa. Bár az ilyen kisebb kategóriák néha elhomályosítják a különbségeket (a felsőkategóriás és olcsó autók megoszthatják ugyanazt a sebességváltót), a gyakorlatban a kódolás jobban stabilizálta a modell viselkedését, mint vártam.

(2) Új jellemzők

Két mérnöki jellemző drámaian javította az eredményeket:

- **brand_model**: a gyártó és a modell összefűzése, lehetővé téve a modell számára, hogy a márkán belüli árváltozást megkülönböztesse (pl. BMW X5 vs. egy olcsóbb BMW modell).
- **car_age** és **mileage_per_year**: származtatott jellemzők, amelyek az értékcsökkenést informatívabban mutatják, mint a kilométeróra vagy a gyártási év önmagában.

Ezek a kombinált jellemzők lettek az egyik legerősebb prediktor mindkét modellben.

(3) Szűrés

Az eloszlási ábrák és kategóriaszámok alapján a kódoló több szűrőt alkalmaz:

- eltávolítja a gyártó és modell kategóriákat, amelyek kevesebb mint 180 előfordulással rendelkeznek,
- eltávolítja a rendkívül olcsó autókat (valószínűleg alkatrészekért eladott) és a rendkívül drága kiugró értékeket,

- eldobja a `levy` mezőt, amely zajt vezetett be anélkül, hogy javított volna az előrejelzéseken (az adó befolyásolja az autó megvásárlásának valószínűségét, nem a piaci árát, és ezért nem tartozik ebbe a modellbe).

A szűrés többet járult hozzá a végső teljesítményhez, mint bármely modellparaméter hangolás.

Sikertelen és elvetett kísérletek

Több modellezési ötlet félrevezető vagy hatástalan volt:

- **A kódolt kategóriák súlyozása a véletlen erdő jellemző fontossága alapján.** Bár elméletileg vonzó, sem a lineáris regresszió, sem a véletlen erdő nem mutatott jelentős változást a súlyok beállításakor. Később megértettem, miért: a lineáris regresszió automatikusan átméretezi az együtthatókat és a véletlen erdő invariáns a bemenetek monoton skálázására.
- **Dimenziócsökkentés alacsony korrelációjú jellemzők eldobásával.** Megpróbáltam eltávolítani minden olyan kategóriát, ahol $|r| < 0,25$ az árhoz képest. Az eredmény rosszabb volt, mert a korreláció önmagában nem képes feltárni a nemlineáris vagy interakciós hatásokat, amelyeket a véletlen erdő kihasznál.
- **Polinomiális jellemzők.** A 2. fok egy kis javulást adott, de a 3. fok és felette numerikusan robbant. Ez várható: a polinomiális jellemzők súlyos multikollinearitást okoznak, hacsak nincsenek gondosan regularizálva.

Ezek a sikertelen utak hasznosak voltak, mert egyértelművé tették, hogy mely összetevők számítanak valóban: tiszta adatok, jól megtervezett kódolók és stabil szűrési szabályok.

3 Modellek

Ebben a projektben két regressziós modellre összpontosítottam, amelyeket a *scikit-learn* gépi tanuló könyvtárral implementáltam [2]:

- Lineáris regresszió (egyszerű, értelmezhető alapvonal)
- Véletlen erdő regresszió (nemlineáris, zajálló)

(Mindkét modell kiértékelésre kerül a 4.3 Modell összehasonlítás részben.)

3.1 Lineáris regresszió [3]

A lineáris regressziót alapmodellként használom egyszerűsége és értelmezhetősége miatt. Fogalmilag a lineáris approximáció közvetlen általánosításának tekinthető magasabb dimenziókban. Ahelyett, hogy egy vonalat illesztenénk az adatokhoz, a modell egy hipersíkot illeszt a többdimenziós jellemzőtérben.

Adott egy bemeneti jellemzővektor $\mathbf{x} \in \mathbb{R}^n$, a modell a célárat $\hat{y}$ a következőképpen jósolja meg:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n,$$

ahol a β_i együtthatókat a előrejelzések és a valódi árak közötti négyzetes hibák átlagának minimalizálásával tanuljuk.

A lineáris regresszió feltételezi a lineáris kapcsolatot a jellemzők és a célváltozó között. Bár ez a feltételezés nyilvánvalóan sérül az autóárazás sok szempontjából, a modell mégis hasznos referenciapontot nyújt. Együtthatói lehetővé teszik az egyes jellemzők hatásának közvetlen megfigyelését a előrejelzett árra, ami segít érvényesíteni az előfeldolgozási és kódolási döntéseket.

A gyakorlatban a lineáris regresszió nehezen modellezte a jellemzők közötti komplex interakciókat, mint például a márka, modell és jármű állapota. Ez a korlátozás motivált egy nemlineáris modell használatát.

3.2 Véletlen erdő regresszió [4]

A véletlen erdő regresszió egy együttes-alapú, nemlineáris regressziós modell, amely döntési fák gyűjteményéből épül fel. Minden fát az adatok és jellemzők véletlen részhalmazán tanítunk, és a végső előrejelzést az összes fa előrejelzéseinek átlagolásával kapjuk.

A lineáris regresszióval ellentétben a véletlen erdők nem feltételeznek lineáris kapcsolatot a bemenetek és a célváltozó között. Ez lehetővé teszi a modell számára, hogy rögzítse a nemlineáris hatásokat és a jellemzők közötti interakciókat, mint például a márka, kilométeróra, gyártási év és motorjellemzők, amelyek fontosak az autóár előrejelzéséhez.

Bár ismerősebb vagyok a lineáris regresszióval és jól értem a viselkedését, a véletlen erdő regresszióval a *nayan2112* kapcsolódó munkája során találkoztam [5]. A legtöbb más online példa, amit találtam, elsősorban lineáris regressziós modellekre támaszkodott. A további vizsgálat azt mutatta, hogy a véletlen erdők jól alkalmazhatók nemlineáris és heterogén adatokra, ezért is vettem bele.

A projekt felénél úgy döntöttem, hogy párhuzamosan dolgozom mind a lineáris, mind a véletlen erdő regressziós modellekkel. Általában a véletlen erdő modellt kezeltem az erősebb jelöltként, de főként összehasonlítás céljából vettem bele, mivel kevésbé ismerem belső működését, mint a lineáris regressziót. Ennek ellenére a lineáris modell végül meglepően jól teljesített kellő adat szűrés és jellemzőkiemelés után.

*

Fejlettebb modellek potenciálisan tovább javíthatnák a teljesítményt, de a választott modellek már bemutatják az előfeldolgozás, jellemzőkiemelés és adatszűrés erős hatását az előrejelzés minőségére.

4 Eredmények

A modellek teljesítményét elsősorban a gyöknégyzetes középhiba (RMSE) és az abszolút relatív hiba átlagával és mediánjával értékeltem. Azt is megvizsgáltam, autónkénti százalékos hibát, hogy megértsem, hogyan viselkedtek a modellek különböző ártartományokban.

4.1 Értékelési metrikák

RMSE volt a fő metrikám. Meghatározása:

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

Ez erősen bünteti a nagy egyedi hibákat. Ez hasznos olyan esetek felderítéséhez, amikor a modell katasztrofális hibákat követ el, de azt is jelenti, hogy egyetlen rossz előrejelzés uralhatja a metrikát. A gyakorlatban azt találtam, hogy az RMSE némileg félrevezető lehet: két különböző viselkedésű modellnek hasonló RMSE értékei lehetnek, és egy magas RMSE-vel rendelkező modellnek mégis lehet alacsony az átlagos relatív hibája.

Ezért kiegészítően értékeltem a modelleket autónkénti relatív hibákkal. Minden autóhoz kiszámítottam az abszolút relatív hibát:

$$e_i = \left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%.$$

Ebből számolom:

- **Átlagos abszolút relatív hiba** (átlagos eltérés autókon),
- **Medián abszolút relatív hiba** (központi mérték, kevésbé érzékeny a kiugró értékekre),
- **Relatív pontosság**, meghatározva:

$$\text{Pontosság} = 100\% - e.$$

A relatív pontosság intuitívabb értelmezést nyújt: azt méri, hogy az előrejelzés milyen közel van a valódi árhoz százalékban kifejezve. Több esetben egy rosszabb RMSE-vel rendelkező modell mégis magasabb relatív pontosságot ért el, különösen a logaritmikus ár szűrés alkalmazása után.

A futási idő nem volt szűk keresztmetszet: mind a lineáris regresszió, mind a véletlen erdő nagyon gyorsan tanult az adathalmaz méretéhez képest, így az időt nem használtam összehasonlítási kritériumként.

4.2 A logaritmikus ár szűrés jelentősége

Központi megfigyelés volt, hogy a rendkívül olcsó autók (100–1000 USD tartomány) és a rendkívül drága autók torzítják a regresszió viselkedését. Ezek a járművek nem úgy viselkednek, mint "normális piaci" árak, és hosszú farokzatot alkottak az eloszlási ábrákon. Az ár logaritmusának használata lehetővé tette a jelentős küszöbértékek azonosítását. (lásd 3. ábra).

- $\ln(p) > 7$ eltávolítja a kb. 1100 USD alatti autókat. Ez kiküszöböli a selejtértékű vagy nem működő járműveket, miközben továbbra is megtart egy széles és valóságos tartományt az alacsony, közepes és magas értékű autókból.
- $\ln(p) > 8$ eltávolítja a kb. 3000 USD alatti autókat. A megmaradt adathalmaz homogénebbé válik, és a közepes és magas értékű járművek dominálnak. Ebben a rendszerben az RMSE gyakran növekszik, mert az abszolút árkülönbségek növekednek, de a relatív pontosság javul, ami konzisztensebb előrejelzéseket jelez egy szűkebb árspektrumon belül.

Ez egy fontos eredmény: a szűrés nem teszi a modellt "csalóvá"; megváltoztatja a probléma definícióját egy realisztikusabb árazási forgatókönyvre.

4.3 Modell összehasonlítás

A 1. táblázat összefoglalja mindkét modell teljesítményét a két szűrési küszöbérték mellett. Az értékek független futtatásokból származnak.

Szűrő	**Modell**	**RMSE**	**Átlagos relatív pontosság**	**Medián relatív pontosság**
$\ln(p) > 7$	Lineáris reg.	≈ 7900	25%	73%
	Véletlen erdő	≈ 4650	73,68 %	90,08 %
$\ln(p) > 8$	Lineáris reg.	≈ 8100[1]	52,9%	75,06 %
	Véletlen erdő	≈ 4550	79,48 %	90,62%

Table 1: Lineáris regresszió és véletlen erdő összehasonlítása két szűrési küszöbérték mellett. Az RMSE abszolút ár egységben van. ($)

4.4 Összefoglaló eredmények

A projekt kezdeti célja volt elérni legalább **80%-os előrejelzési pontosságot** lineáris regresszió használatával. Ez a cél nem érhető el következetesen az értékelt metrikák egyike alatt sem. Azonban közelebbről vizsgálva látható, hogy ez az eredmény erősen befolyásolt a hibaeloszlástól, nem pedig szisztematikus modellhibától. Míg az **átlagos relatív pontosság** alacsony volt, a **medián relatív pontosság** az alacsony-közepes 70%-os tartományban maradt, ami azt jelzi, hogy a legtöbb előrejelzés meglehetősen pontos volt.

Az átlag és a medián pontosság közötti eltérés egy erősen ferde hibaeloszlást tár fel: amikor a lineáris regresszió hibáz, néha nagyon nagy (néha meghaladja a 100%-ot), de az előrejelzések többsége egy kis hibatartományba esik,

[1] Az RMSE növekedése a $\ln(p) > 8$ esetén a lineáris regressziónál azért következik be, mert a megmaradt autók szélesebb abszolút ártartományt fednek le, nem azért, mert a modell rosszabbul teljesít.

gyakran 10% alatt, és sok esetben 5% alatt. Kis számú szélsőséges kiugró érték tehát uralja az átlagos metrikát és torzítja az összesített értékelést.

A véletlen erdő regresszió következetesen felülmúlta a lineáris regressziót mind az RMSE, mind a relatív pontosság tekintetében. A $\ln(p) > 7$ szűréssel a legjobb egyensúlyt érte el az adathalmaz mérete és az előrejelzés stabilitása között, míg a $\ln(p) > 8$ tovább javította a relatív pontosságot magasabb RMSE árán, különösen a lineáris regressziónál. A **legjobb teljesítményt** nyújtó konfiguráció a **véletlen erdő modell** volt $\ln(p) > 7$ mellett, amely körülbelül 4600–5000 közötti RMSE értékeket ért el, miközben magas relatív pontosságot tartott fenn a legkevésbé módosított adathalmazon.

Ezek az eredmények azt sugallják, hogy további előszűrés vagy korlátozott emberi felügyelet, például a súlyosan alul- vagy túlértékelt hirdetések eltávolítása, tovább javíthatná a lineáris regresszió teljesítményét, és potenciálisan közelebb hozhatná az eredeti pontossági célhoz. Hasonló megközelítéseket használtak online példákban, mint Zabihullah jegyzetfüzete [6] és Nayan Shreemen Pale munkája [7]. Mindkettő elérte a 80%+ pontosságot lineáris regresszióval, de csak sokkal erősebb szűréssel, mint amit én elfogadhatónak tartottam. Zabihullah esetében ezt kombinálták több mint egy tucat további adathalmazzal, amelyek közül sok sokkal gazdagabb és részletesebb autóattribútumokat tartalmazott, mint a Kaggle [1] adathalmaz, amit én használtam. Bár hatékony a pontosság növelésére, véleményem szerint ez a szintű szűrés és adathalmaz keverése sérti az eredeti probléma integritását, és az előrejelzési feladatot egy sokkal könnyebb, homogénebb járműhalmazra szűkíti.

5 Jövőbeli munkák

A jövőbeli munkák között szerepel a modell üzembe helyezése autóárazó API-ként vagy webalapú alkalmazásként. Elkezdtem egy ilyen alkalmazás fejlesztését, de az üzembe helyezés akkorra halasztódik, amíg a modell pontossága javul.

További javulások származhatnak szigorúbb adatszűrésből, külön modellek képzéséből az alacsony, közepes és magas értékű autókra, és több adathalmaz kombinálásából az általánosítás javítására. A meglévő objektumorientált struktúra is kiterjeszthető jobb skálázhatóság és újrafelhasználás érdekében. Végül, fejlettebb regressziós modellek is vizsgálhatók, bár ez a projekt szándékosan a lineáris regresszióra és véletlen erdő módszerekre összpontosított.

References

[1] Deep Contractor. Car price prediction challenge. `https://www.kaggle.com/datasets/deepcontractor/car-price-prediction-challenge`, 2022. [Online; accessed 12/14/2025].

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[3] Scikit Learn. Linearregression. `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html`, 2025. [Online; accessed 12/14/2025].

[4] Scikit Learn. Randomforestregressor. `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html`, 2025. [Online; accessed 12/14/2025].

[5] nayan2112. Car-price-prediction. `https://github.com/nayan2112/Car-Price-Prediction`, 2021. [Online; accessed 12/14/2025].

[6] Zabihullah1. Car price prediction. `https://www.kaggle.com/code/zabihullah18/car-price-prediction/notebook`, 2024. [Online; accessed 12/14/2025].

[7] Nayanshree Menpale. linearregression_carpriceprediction. `https://www.kaggle.com/code/nayanshreemenpale/linearregression-carpriceprediction/notebook`, 2025. [Online; accessed 12/14/2025].