Zadanie č.2

Stromy, stroje, hlasovania a redukcia dimenzie

Autor: Adrián Somor

Úloha č.1: Trénovanie stromu, lesu a SVM

Cieľom úlohy bolo spracovať a vyčistiť daný dátový súbor odstránením odchýliek, riešením chýbajúcich hodnôt a kódovaním nečíselných stĺpcov. Po príprave dát bolo potrebné vytvoriť vstupné a výstupné dátové sady, rozdeliť ich na trénovacie a testovacie množiny a následne dáta škálovať. Potom som pomocou týchto dát natrénoval jednoduchý strom, les a SVM. Nakoniec som vyhodnotil natrénované modely na trénovacej aj testovacej množine.

Predspracovanie dát

Stĺpce, ktoré nie sú nevyhnutné pre ďalšie spracovanie alebo modelovanie boli vynechané.

ID je jedinečný identifikátor, preto som ho odstránil. Stĺpce *Manufacturer* a *Model* po OHE vytvoria veľké množstvo stĺpcov, preto boli odstránené. Stĺpcu *Levy* chýba veľa hodnôt, preto som sa rozhodol ho tiež odstrániť.

Odstránenie outlierov

Outlieri môžu skresliť výsledky našej analýzy a viesť k nesprávnym záverom.

Prod. year: rozsah od 1992
Price: Cena v rozsahu od 500 do 100000
Engine volume: Rozsah 1 az 10
Mileage: Uprava hodnot na spojit´a ˇc´ıseln´a hodnota a rozsah od 0 do 500000
Cylinders: Rozsah 4 az 12

Z datasetu boli taktiež odstránené aj duplikáty

Kódovanie nečíselných stĺpcov

Stĺpece *Color, Category, Fuel type, Gear box type, Drive wheels, Doors sú* kategorické a nemajú prirodzené usporiadanie. Preto som sa rozhodol použiť "one hot encoding" na tieto stĺpce. Okrem toho som stĺpece *Leather interior, Left wheel, Turbo engine*, ktorý je kategorický, ale môže byť reprezentovaný číselne (pravda/hodnota nepravdy), premenil na int formát.

Vytvorenie vstupných a výstupných dátových sád

Najprv som oddelil stĺpce reprezentujúce cenu (výstupnú hodnotu) od zvyšku datasetu.

```
X = df.drop(columns=['Price'])
y = df['Price']
```

Rozdelenie dát na trénovacie, validačné a testovacie množiny

Potom som rozdelil dáta na trénovacie, validačné a testovacie množiny v pomere 8:1:1

X_train, X_test, y_train, y_test = train_test_split(X, y, shuffle=True, test_size=0.2, random_state=69)
X_valid, X_test, y_valid, y_test = train_test_split(X_test, y_test, shuffle=True, test_size=0.5, random_state=69)

Škálovanie dát

Pred trénovaním modelu je dôležité, aby mali všetky features rovnaký vplyv na model. Jedným zo spôsobov, ako to dosiahnuť, je škálovanie alebo normalizácia (toto platí pre SVM).

Škáloval som dáta pomocou metódy StandardScaler, ktorá transformuje dáta tak, aby mali priemernú hodnotu 0 a štandardnú odchýlku 1. Následne som premenil numpy polia na pandas DataFrames.

Trénovanie rozhodovacieho stromu

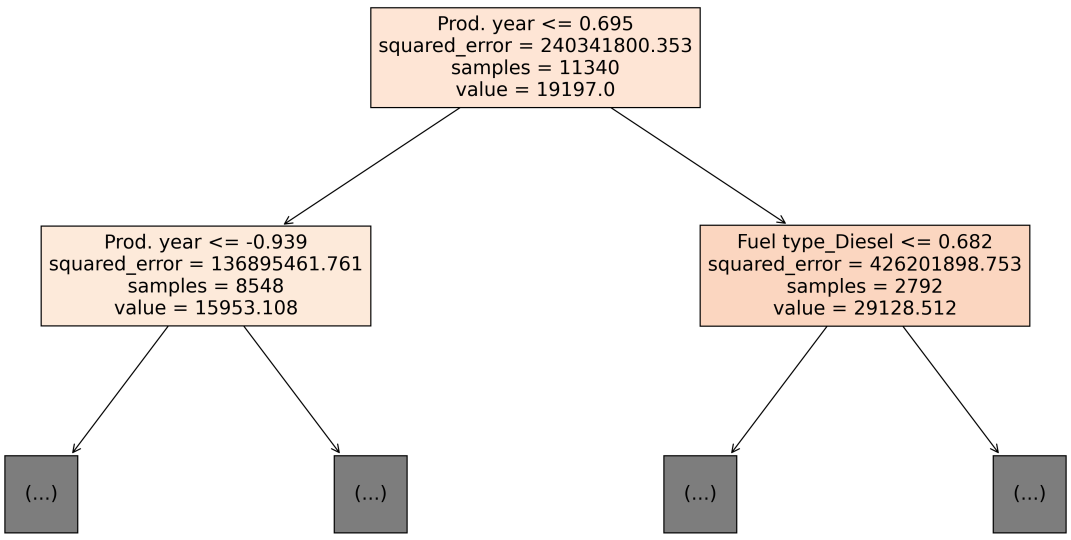
Pre analýzu a regresiu som sa rozhodol použiť rozhodovací strom. K tomu som použil triedu DecisionTreeRegressor z knižnice sklearn.

Pri definovaní modelu som nastavil nasledovné parametre:

- **max_depth=8:** Špecifikoval som maximálnu hĺbku stromu
- **random_state=42:** Zabezpečuje, že výsledky budú konzistentné pri opakovanom spustení. To je užitočné pre reprodukovateľnosť experimentu.

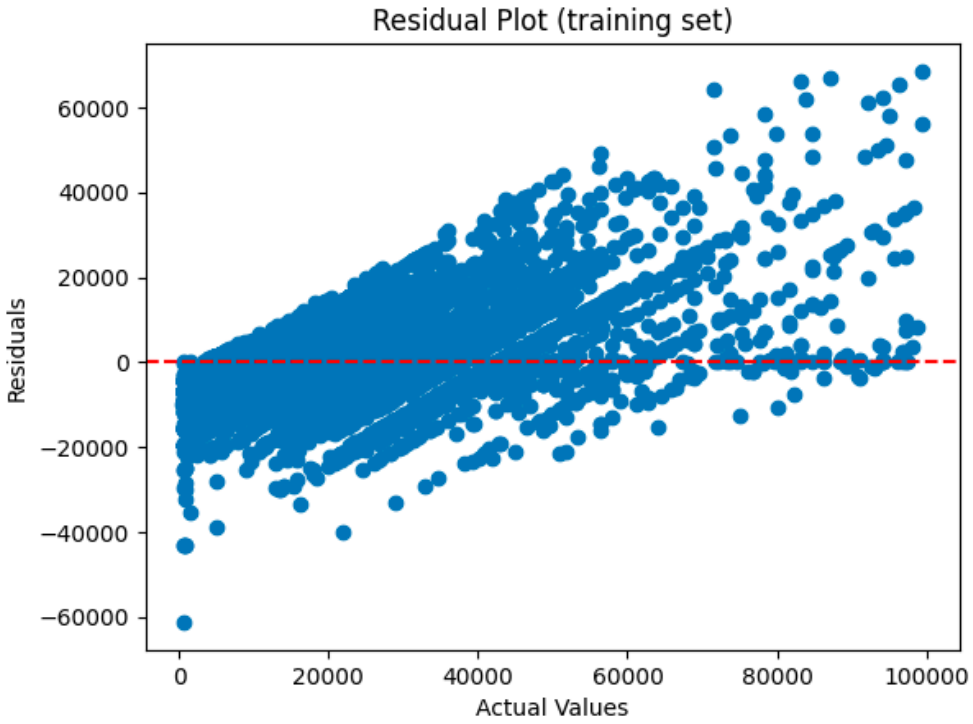
Následne som model natrénoval na trénovacích dátach pomocou metódy fit.

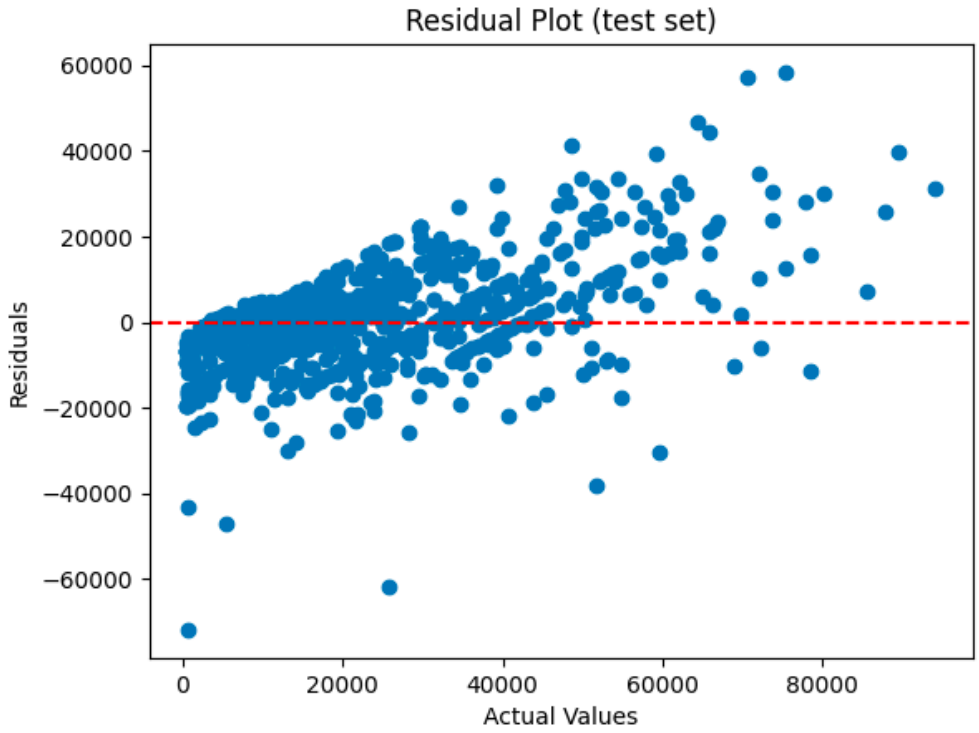
Vizualizácia (časti) rozhodovacieho stromu



Vyhodnotenie rozhodovacieho stromu

Train MSE: 73977181.89340228	Test MSE: 92176367.76433286
Train R2 Score: 0.692200100920269	Test R2 Score: 0.5929146758114427





Trénovanie lesu

K tomu som použil triedu RandomForestRegressor z knižnice sklearn.

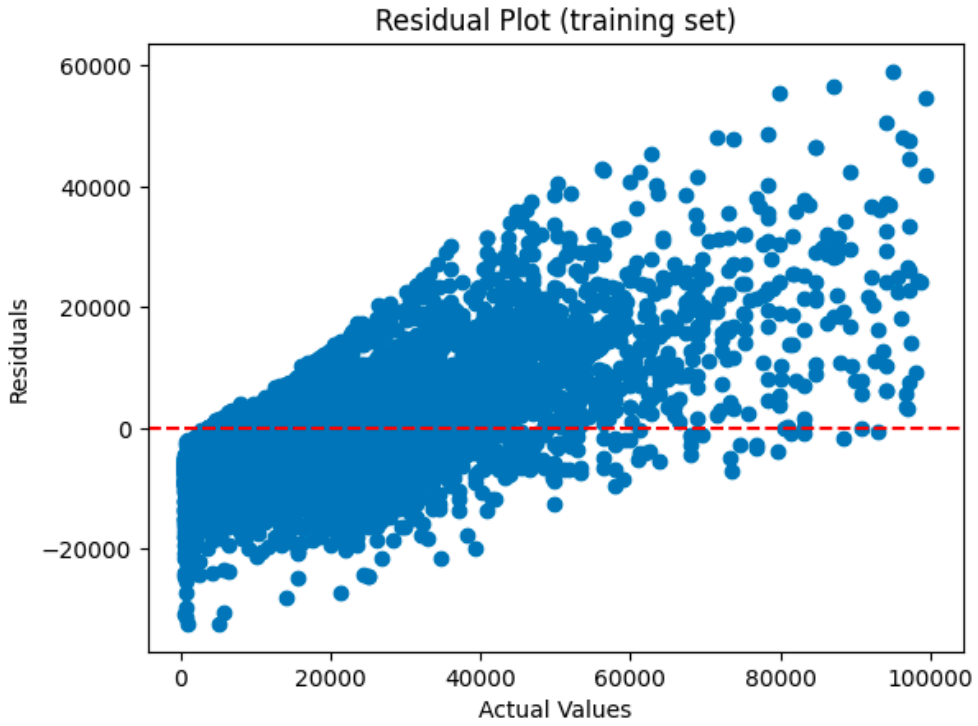
Pri definovaní modelu som nastavil nasledovné parametre:

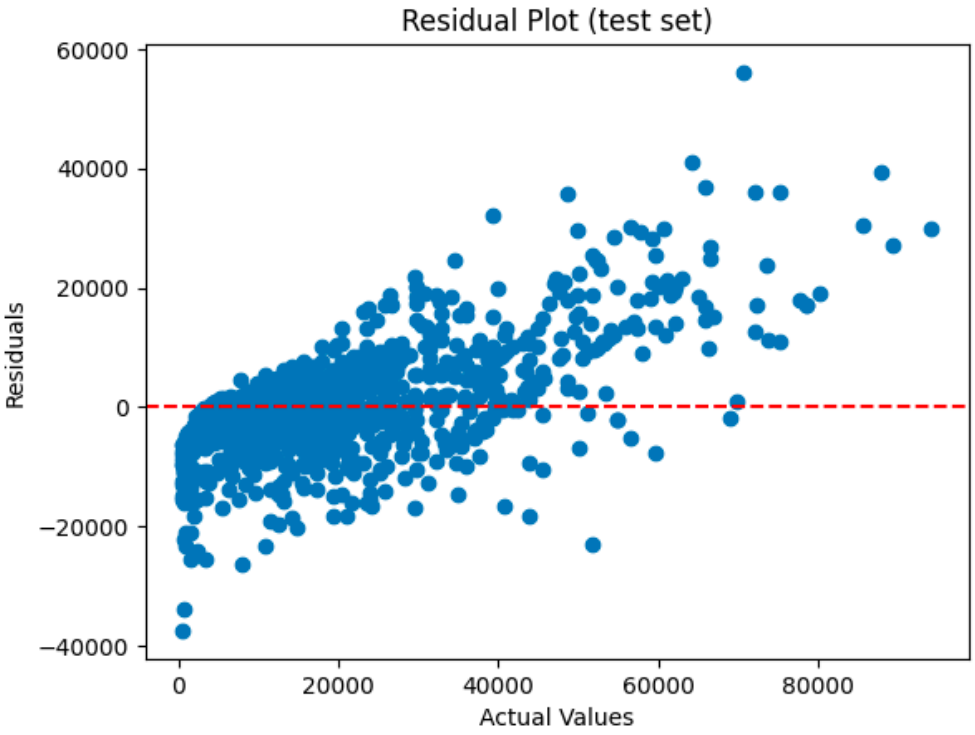
max_depth=8: Špecifikoval som maximálnu hĺbku stromu
random_state=42: Zabezpečuje, že výsledky budú konzistentné pri opakovanom spustení. To je užitočné pre reprodukovateľnosť experimentu.
n_estimators=100: Špecifikuje počet stromov v lese

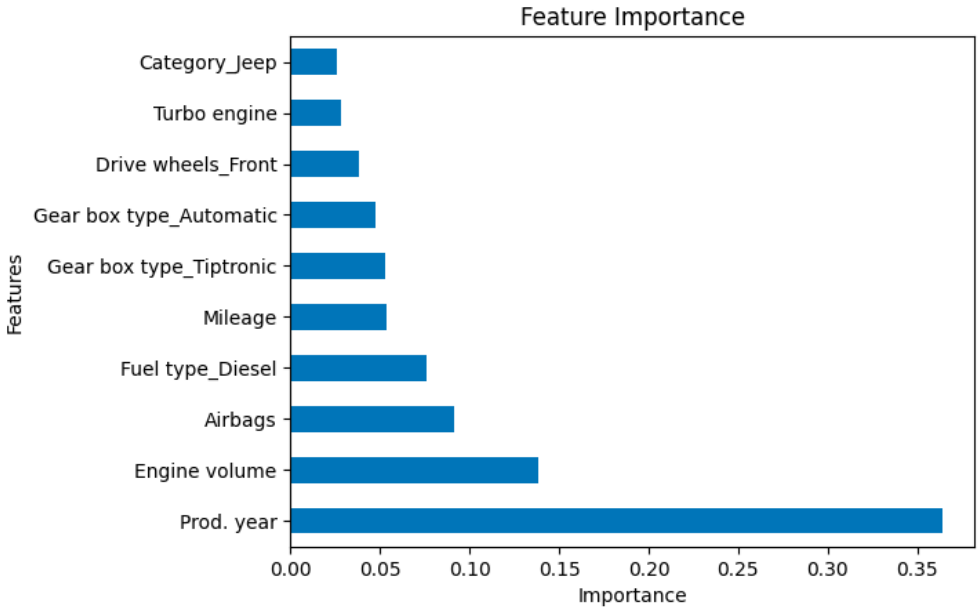
Následne som model natrénoval na trénovacích dátach pomocou metódy fit.

Vyhodnotenie lesu

Train MSE: 59520648.53803489	Test MSE: 69769384.52451688
Train R2 Score: 0.7523499930077601	Test R2 Score: 0.691872296484771




Residual Plot (test set)
Residuals
Actual Values
60000
40000
20000
0
−20000
−40000
0
20000
40000
60000
80000


Feature Importance
Features
Importance
Category_Jeep
Turbo engine
Drive wheels_Front
Gear box type_Automatic
Gear box type_Tiptronic
Mileage
Fuel type_Diesel
Airbags
Engine volume
Prod. year
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35

Trénovanie SVM

K tomu som použil triedu SVR z knižnice sklearn.

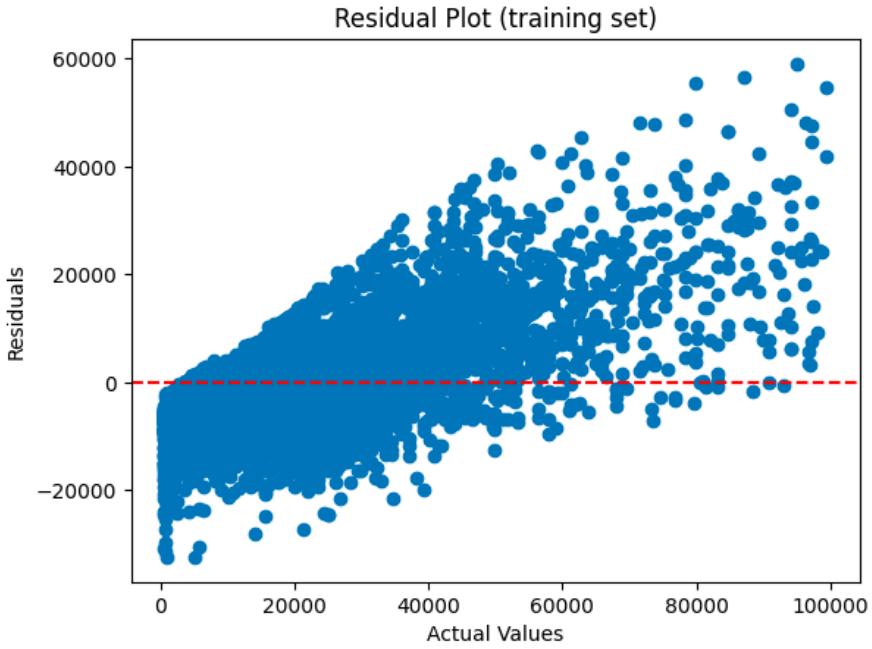
Pri definovaní modelu som nastavil nasledovné parametre:

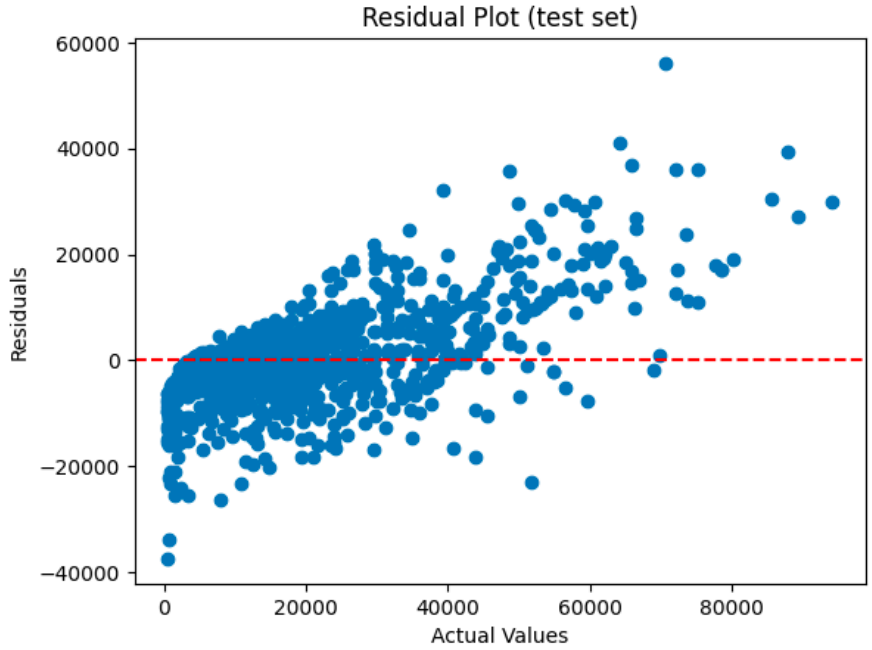
kernel='linear': Špecifikoval som kernel ‘linear’ vykonáva regresiu podporovacích vektorov pomocou lineárnej hyperroviny na modelovanie vzťahu medzi nezávislými premennými a spojitou závislou premennou.

Následne som model natrénoval na trénovacích dátach pomocou metódy fit.

Vyhodnotenie SVM

Train MSE: 181857684.0861877	Test MSE: 172694376.04258895
Train R2 Score: 0.24333726460058047	Test R2 Score: 0.23731702862736737





Porovnanie modelov

Rozhodovací Strom: Na obrázku pre rozhodovací strom vidíme, že reziduály majú trend zvyšovať sa s rastúcimi skutočnými hodnotami. To naznačuje, že model má problémy s predpoveďou presnejších hodnôt pre vyššie hodnoty cieľovej premennej.

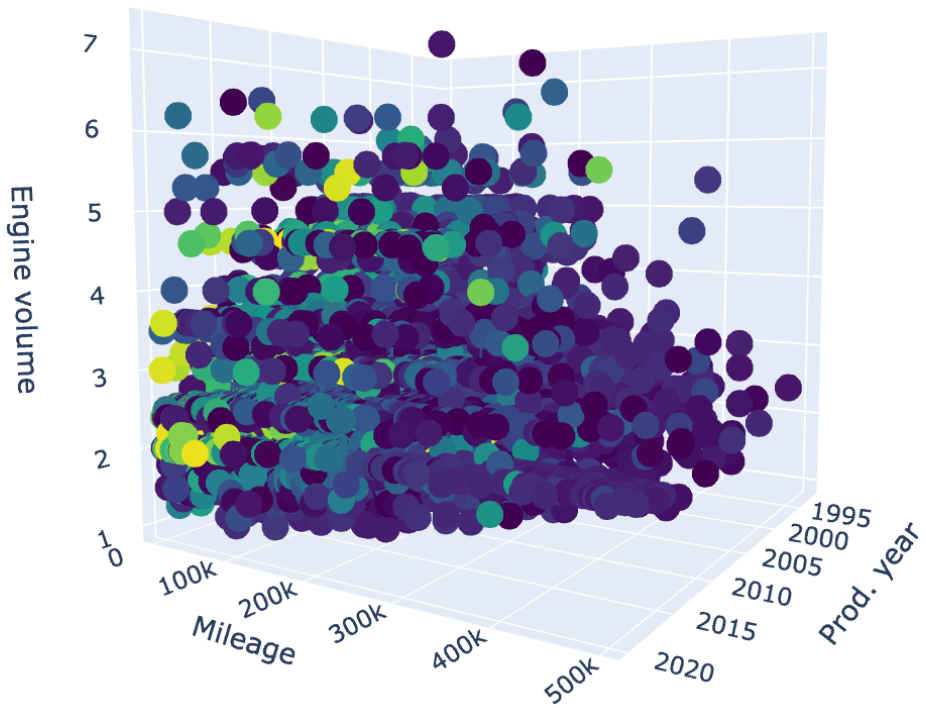
Náhodný Les: Na obrázku, ktorý predstavuje náhodný les, je vidieť, že rozptyl reziduálov je menší ako pri rozhodovacom strome, najmä v nižšom rozsahu skutočných hodnôt. Reziduály však stále ukazujú trend: sú nižšie pre menšie skutočné hodnoty a vyššie pre väčšie skutočné hodnoty. Tento model teda zrejme lepšie zovšeobecňuje ako rozhodovací strom, ale stále by mohol byť vylepšený.

SVM: Na obrázku, ktorý reprezentuje SVM, je distribúcia reziduálov pomerne rovnomerná okolo nulovej čiary, čo je pozitívny znak. Napriek tomu je tu stále viditeľný určitý trend, kde reziduá majú tendenciu byť pozitívne (nad nulovou čiarou) pre vyššie skutočné hodnoty. To môže indikovať, že model systematicky podhodnocuje vyššie hodnoty.

Úloha č.2: Redukcia dimenzie

Cieľom tejto úlohy je aplikovať redukciu dimenzií na dataset s cieľom vizualizovať kľúčové prvky a vylepšiť prediktívny model. Vyberiem tri prvky pre 3D bodový graf, použijem PCA alebo UMAP na redukciu dát na tri dimenzie po normalizácii a porovnám tieto vizualizácie. Potom určím najlepšiu podmnožinu prvkov cez koreláciu, dôležitosť prvkov z ansámblového modelu a varianciu PCA, znovu naškolím optimálny model na týchto podmnožinách a zhodnotím zlepšenia výkonu pomocou MSE, RMSE, R^2 a reziduálnej analýzy.

Scatter plot #1



Mileage vs. Prod. year: Vyšší počet najazdených kilometrov je zreteľnejší u starších vozidiel, čo je očakávané. Novšie modely (bližšie k roku 2020) sa zdajú mať nižší počet najazdených kilometrov, čo zodpovedá ich nedávnejšiemu vstupu na trh.

Scatter plot #2

Porovnanie scatter plotov

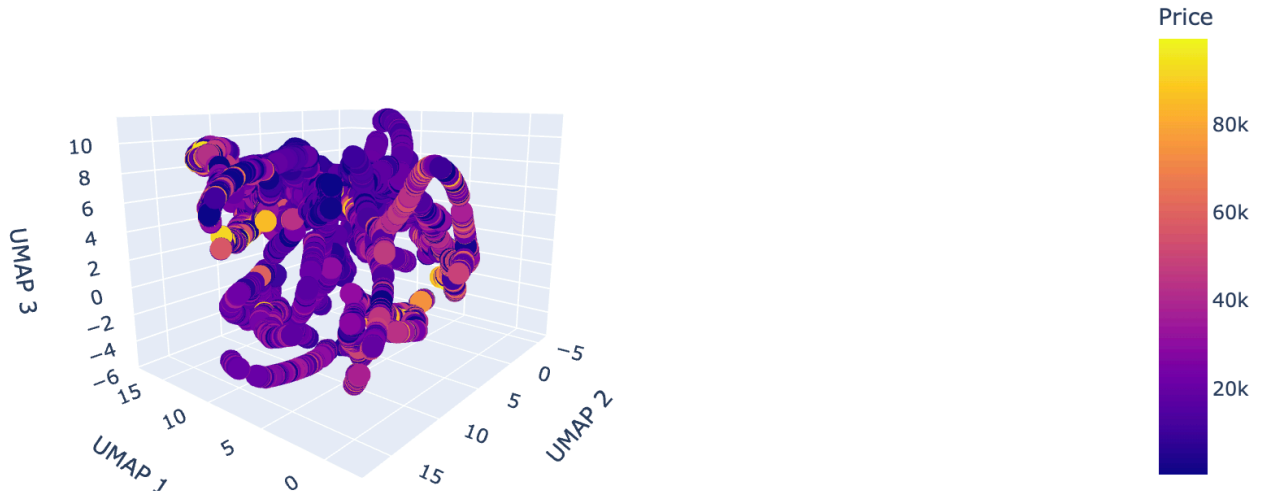
Prvý graf má za cieľ prezentovať priame vzťahy medzi konkrétnymi, interpretabilnými premennými.

Druhý graf je určený na vizualizáciu výsledku techniky redukcie dimenzií, ktorá abstrahuje od pôvodných vlastností do hlavných komponentov.

Interpretovateľnosť

Osi v grafe PCA nezodpovedajú priamo pôvodným vlastnostiam a na interpretáciu v kontexte pôvodných údajov je potrebné mať odborné vedomosti.

Naopak, prvý graf zobrazuje aktuálne vlastnosti, ktoré je možné pochopiť bez nutnosti porozumenia transformačnému procesu.



Pri pohľade na rozloženie bodov pozorujeme zhluky alebo skupiny s rôznymi hustotami a rozdielmi medzi nimi. Tieto zhluky by mohli naznačovať prirodzené skupiny v pôvodných dátach s vysokou dimenzionalitou, ktoré sú teraz viditeľné po redukcii dimenzií.h

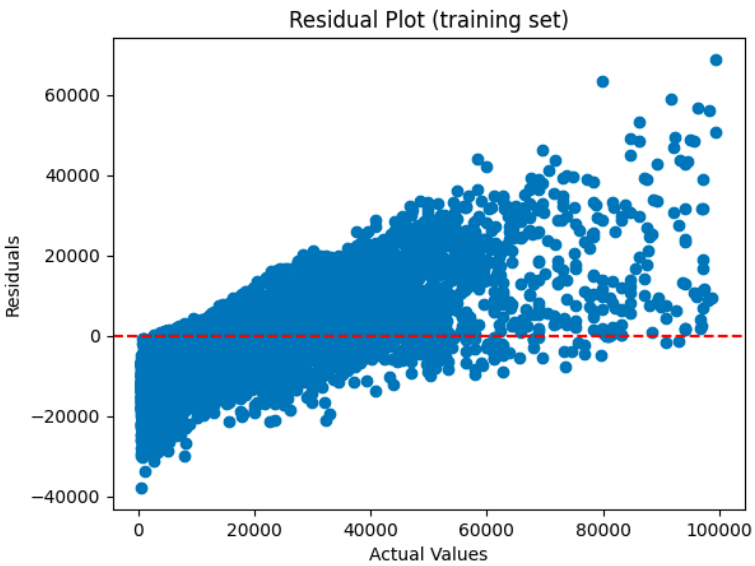
Úloha č.3: Trénovanie najlepšieho modelu

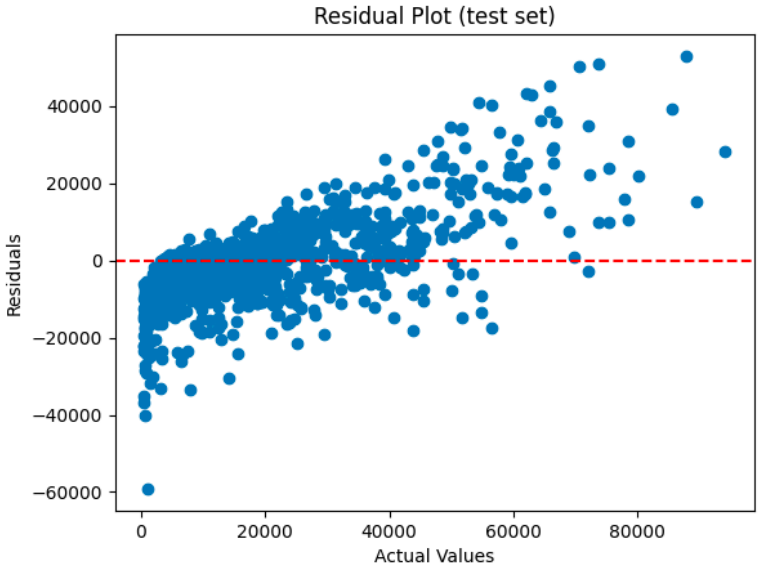
Podľa korelačnej matice

Top 10 Positive Correlations with Price:
Prod. year 0.411976
Category_Jeep 0.298132
Fuel type_Diesel 0.243400
Leather interior 0.220339
Left wheel 0.182578
Engine volume 0.168520
Gear box type_Tiptronic 0.167531
Turbo engine 0.148471
Cylinders 0.091097
Drive wheels_4x4 0.082592

Top 10 Negative Correlations with Price:
Mileage -0.228300
Gear box type_Manual -0.165252
Category_Hatchback -0.158923
Category_Sedan -0.151534
Fuel type_CNG -0.121046
Fuel type_Hybrid -0.094717
Color_Silver -0.092871
Category_Goods wagon -0.072719
Color_Green -0.071864
Fuel type_Petrol -0.068404

```
selected_features = [
    'Prod. year', 'Category_Jeep', 'Fuel type_Diesel', 'Leather interior', 'Mileage',
    'Left wheel', 'Engine volume', 'Gear box type_Tiptronic', 'Turbo engine',
    'Gear box type_Manual', 'Category_Hatchback', 'Category_Sedan', 'Fuel type_CNG'
]
```



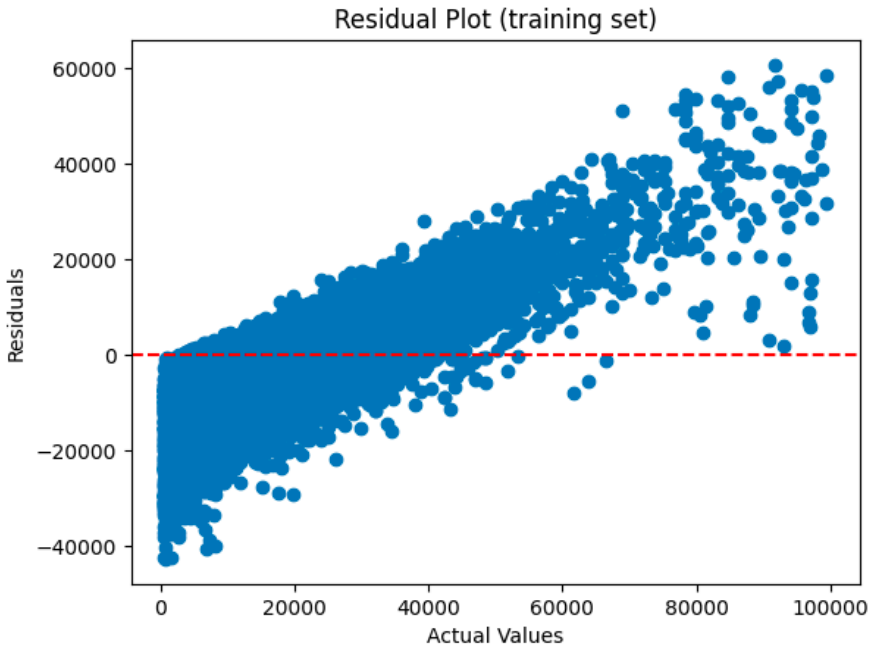


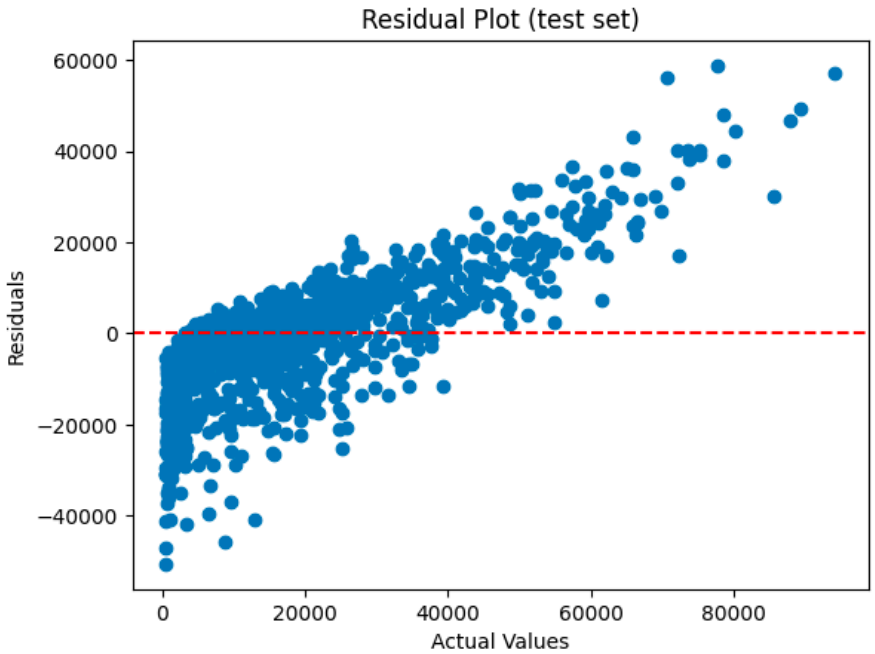
Train MSE: 69998804.70275164	Test MSE: 98488342.91680422
Train R2 Score: 0.7087530982965125	Test R2 Score: 0.5650386321623441

Podl'a dôležitosti príznakov z ensemble modelu

```
selected_features = [
    'Prod. year', 'Engine volume', 'Mileage'
]
```

Train MSE: 107972629.44330116	Test MSE: 147199724.73774755
Train R2 Score: 0.5507538460452588	Test R2 Score: 0.34991094660469835

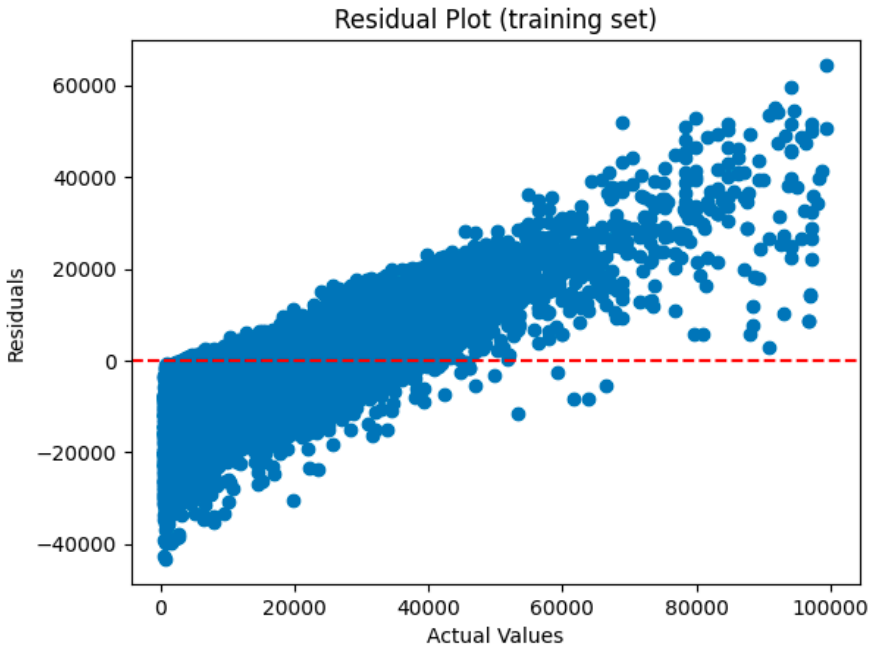


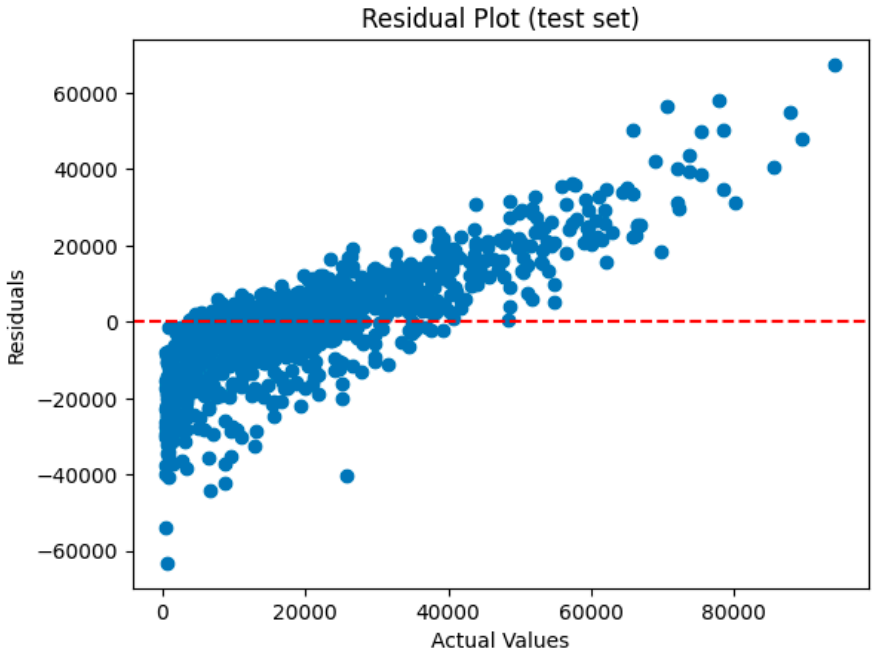


Podl'a variancie pomocou PCA

Threshold pre varianciu som nastavil na 0.9

Train MSE: 103944519.96081464	Test MSE: 160001547.44352385
Train R2 Score: 0.5675137666107384	Test R2 Score: 0.2933733082404981





Bonus: Trénovanie neurónovej siete

Dáta boli “čistené” rovnako ako pri predošlých úlohách, nastavenia neurónovej siete boli nasledovné:

Pre túto úlohu som použil MLPRegressor z knižnice sklearn

hidden_layer_sizes=(256, 256, 256, 256, 256): Sieť pozostáva z piatich skrytých vrstiev, každá s 256 neurónmi.

activation='relu': Aktivačná funkcia pre skryté vrstvy je rectified linear unit (ReLU). Táto funkcia zavádza nelinearitu do modelu, čo mu umožňuje učiť sa zložitejšie funkcie.

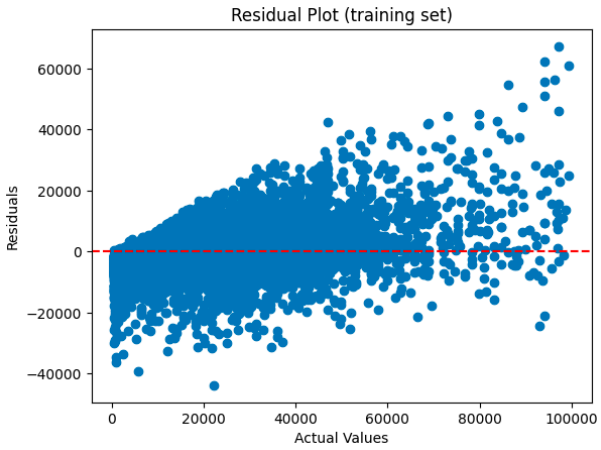
solver='adam': Riešič pre optimalizáciu váh je 'adam', čo je stochastický gradientový optimalizátor.

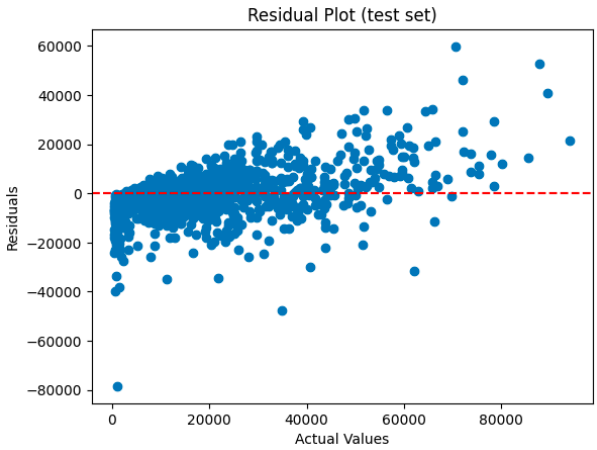
random_state=42: Seed používané generátorom náhodných čísel je 42. To zabezpečuje reprodukovateľnosť výsledkov, pretože rovnaké semeno vždy vyprodukuje rovnakú postupnosť náhodných čísel pri každom spustení kódu.

early_stopping=True: Skoré zastavenie sa používa na predchádzanie pretrénovaniu. Tréning sa zastaví, ak sa skóre validácie nezlepšuje počas n_iter_no_change po sebe idúcich epoch.

n_iter_no_change=10: Tento parameter pracuje v tandeme s early_stopping. Nastavuje počet iterácií bez zlepšenia na validačnom skóre, ktoré sa čaká pred skorým zastavením tréningu. Tu je nastavený na 10.

learning_rate='adaptive': Plán učenia je adaptívny, čo znamená, že učiaci sa kurz sa zníži, ak sa tréning nezlepšuje po určitý počet iterácií. To zabezpečuje, že model konverguje, aj keď začína s príliš vysokým kurzom učenia.





Train MSE: 48358055.6135805	Test MSE: 76957213.8241748
Train R2 Score: 0.7987946518562963	Test R2 Score: 0.6601281532555183

Bonus: Clustering

Pri implementácii algoritmu zhlukovania som najprv inicializoval KMeans algoritmus z knižnice scikit-learn, nastavujúc požadovaný počet zhlukov na tri. Pre prípravu dát na zhlukovanie som vybral číselné prvky—'*Prod. year*', '*Engine volume*', '*Mileage*', '*Cylinders*' a '*Airbags*'—a binárne prvky—'*Leather interior*' a '*Turbo engine*'—z DataFrame `df`. Tieto súbory prvkov som spojil do jediného DataFrame s názvom `features_for_clustering` pomocou `pd.concat`, čím som zaistil, že sú zarovnané podľa stĺpcov. Na tomto kombinovanom DataFrame aplikujem metódu `fit_predict` inštancie KMeans na vykonanie zhlukovania, a výsledné priradenie zhlukov potom pridám do `features_for_clustering` ako nový stĺpec. Na vizualizáciu využívam funkciu `px.scatter_3d` z Plotly na vygenerovanie 3D bodového grafu zhlukovaných dát, kde '*Prod. year*', '*Engine volume*' a '*Mileage*' sú osi a príslušnosť k zhluku určuje farbu bodov.