

I-SUNS: Zadanie č. 2

Richard Körösi (111313)

Deadline: 16.11.2023

1 Prvá časť

1.1 Načítanie a spracovanie dát v datasete

Dataset, ktorý nám bol poskytnutý obsahoval aj dátu, ktoré sme museli pred ich samotným využitím upraviť. Presnejšie išlo o odstránenie identifikátorov, null hodnôt, duplikátov a outlierov, zároveň sme mali za úlohu správne spracovať textové hodnoty.

1.1.1 Odstránenie identifikátorov

Dataset obsahoval stĺpec ID, ktorý sme odstránili.

```
1 def handleIdentifierColumns(dframe):
2     dframe = dframe.drop(['ID'], axis=1)
3     return dframe
```

1.1.2 Odstránenie nedôležitých dát

V tejto časti sme sa zamerali na dátu, ktoré sme považovali bud za nedôležité, alebo na dátu, ktoré by po spracovaní z textovej hodnoty vytvorili príliš veľa stĺpcov.

```
1 def handleUselessColumns(dframe):
2     dframe = dframe.drop(
3         ['Left wheel', 'Color', 'Model', 'Manufacturer'], axis=1)
4     return dframe
```

Z kódu vieme teda vyčítať že sme odstránili 4 stĺpce: 'Left wheel', 'Color', 'Model' a 'Manufacturer'.

1.1.3 Odstránenie null hodnôt

Dataset obsahoval stĺpec 'Levy', v ktorom malo pomerne veľa riadkov (5818) nezadanú hodnotu, túto hodnotu sme považovali za null hodnotu a vzhľadom na počet týchto 'null' hodnôt sme sa rozhodli daný stĺpec odstrániť. Pri ostatných stĺpcoch sme žiadne null hodnoty neobjavili.

```
1 def handleNullValues(dframe):
2     dframe = dframe.drop(['Levy'], axis=1)
3     return dframe
```

1.1.4 Odstránenie duplikátov

Následne sme odstránili duplikáty v datasete. Pred odstránením duplikátov mal dataset 19237 riadkov, po ich odstránení mal 15685 riadkov.

```
1 def handleDuplicateValues(dframe):
2     dframe = dframe.drop_duplicates()
3     return dframe
```

1.1.5 Spracovanie textových hodnôt

V tejto časti sme sa venovali textovým hodnotám ako zobrazuje obrázok č.1. Textové dáta sme si spracovali pomocou dvoch funkcií, v prvej funkcií *handleTextToNumeric* sme spracovali stlpce 'Mileage' a 'Leather interior'.

#	Price	Prod. year	Category	Leather interior	Fuel type	Engine volume	Mileage	Cylinders	Gear box type	Drive wheels	Doors	Airbags	Turbo engine
13328.0...	2010	Jeep	Yes	Hybrid	3.50000	186005 km	5	Automatic	4x4	4-5	12	False	
16621.0...	2011	Jeep	No	Petrol	3.00000	192000 km	5	Tiptronic	4x4	4-5	8	False	
8467.00...	2006	Hatchback	No	Petrol	1.30000	200000 km	4	Variator	Front	4-5	2	False	
3607.00...	2011	Jeep	Yes	Hybrid	2.50000	168966 km	4	Automatic	4x4	4-5	0	False	
11726.0...	2014	Hatchback	Yes	Petrol	1.30000	91901 km	4	Automatic	Front	4-5	4	False	
39493.0...	2016	Jeep	Yes	Diesel	2.00000	160931 km	4	Automatic	Front	4-5	4	False	
1803.00...	2010	Hatchback	Yes	Hybrid	1.80000	258909 km	4	Automatic	Front	4-5	12	False	
549.00...	2013	Sedan	Yes	Petrol	2.40000	216118 km	4	Automatic	Front	4-5	12	False	
1098.00...	2014	Sedan	Yes	Hybrid	2.50000	398069 km	4	Automatic	Front	4-5	12	False	
26657.0...	2007	Jeep	Yes	Petrol	3.50000	128500 km	5	Automatic	4x4	4-5	12	False	
941.000...	2014	Sedan	Yes	Diesel	3.50000	184467 km	5	Automatic	Rear	4-5	12	False	
8781.00...	1999	Microbus	No	CNG	4.00000	0 km	3	Manual	Rear	2-3	0	False	
3000.00...	1997	Goods wagon	No	CNG	1.60000	350000 km	4	Manual	Front	4-5	4	False	
1019.00...	2013	Jeep	Yes	Hybrid	3.50000	138038 km	5	Automatic	Front	4-5	12	False	
59464.0...	2016	Jeep	Yes	Diesel	2.00000	76000 km	4	Automatic	Front	4-5	4	False	

Figure 1: Dáta pred spracovaním

```
1 def handleTextToNumericBool(dframe):
2     dframe['Mileage'] = dframe['Mileage'].str.split(' ').str[0].\
3         astype(float)
4     dframe['Leather interior'] = dframe['Leather interior'].map(
5         {'Yes': True, 'No': False})
6     return dframe
```

Price	Prod. year	Category	Leather interior	Fuel type	Engine volume	Mileage	Cylinders	Gear box type	Drive wheels	Doors	Airbags	Turbo engine
13328.0...	2010	Jeep	True	Hybrid	3.50000	186005.00000	6	Automatic	4x4	4-5	12	False
16621.0...	2011	Jeep	False	Petrol	3.00000	192000.00000	6	Tiptronic	4x4	4-5	8	False
8467.00...	2006	Hatchback	False	Petrol	1.30000	200000.00000	4	Variator	Front	4-5	2	False
3607.00...	2011	Jeep	True	Hybrid	2.50000	168966.00000	4	Automatic	4x4	4-5	0	False
11726.0...	2014	Hatchback	True	Petrol	1.30000	91901.00000	4	Automatic	Front	4-5	4	False
39493.0...	2016	Jeep	True	Diesel	2.00000	160931.00000	4	Automatic	Front	4-5	4	False
1803.00...	2010	Hatchback	True	Hybrid	1.80000	258909.00000	4	Automatic	Front	4-5	12	False
549.00...	2013	Sedan	True	Petrol	2.40000	216118.00000	4	Automatic	Front	4-5	12	False
1098.00...	2014	Sedan	True	Hybrid	2.50000	398069.00000	4	Automatic	Front	4-5	12	False
26657.0...	2007	Jeep	True	Petrol	3.50000	128500.00000	6	Automatic	4x4	4-5	12	False
941.00...	2014	Sedan	True	Diesel	3.50000	184467.00000	6	Automatic	Rear	4-5	12	False
8781.00...	1999	Microbus	False	CNG	4.00000	0.00000	8	Manual	Rear	2-3	0	False
3000.00...	1997	Goods wagon	False	CNG	1.60000	350000.00000	4	Manual	Front	4-5	4	False
1019.00...	2013	Jeep	True	Hybrid	3.50000	138038.00000	6	Automatic	Front	4-5	12	False
59464.0...	2016	Jeep	True	Diesel	2.00000	76000.00000	4	Automatic	Front	4-5	4	False

Figure 2: Dáta po spracovaní

V druhej funkcií *handleCategoricalValues* sme sa zamerali na dátu, na ktoré sme chceli použiť Dummy encoding, poprípade Label encoding. Label encoding sme sa rozhodli aplikovať na stĺpec 'Doors', keďže dané dátu vieme zoradiť, čím viacej dverí, tým väčšia Label hodnota (2-3 reprezentuje číslo 1, 4-5 reprezentuje číslo 2 atď.). Ostatné stĺpce enkódovali pomocou Dummy encodingu.

```

1 def handleCategoricalValues(dframe):
2     dframe['Doors'] = dframe['Doors'].map({'2-3': 1, '4-5': 2, ' >5': 3})
3     dframe = pd.get_dummies(dframe, columns=['Category'], prefix='Category_', prefix_sep='')
4     dframe = pd.get_dummies(dframe, columns=['Fuel type'], prefix='FuelType_', prefix_sep='')
5     dframe = pd.get_dummies(dframe, columns=['Gear box type'], prefix='Gearbox_', prefix_sep='')
6     dframe = pd.get_dummies(dframe, columns=['Drive wheels'], prefix='Drive_', prefix_sep='')
7     return dframe

```

	Doors	Airbags	Turbo engine	Category_Cabriolet	Category_Coupe	Category_Goods wagon	Category_Hatchback
1	8	False	True	False	False	False	False
1	8	False	False	False	False	True	False
1	8	False	False	True	False	False	False
1	4	False	False	True	False	False	False
1	12	True	False	False	False	False	False
1	6	False	False	True	False	False	False
1	12	False	False	True	False	False	False
1	2	False	False	False	False	False	False
1	6	False	False	False	False	False	False
1	6	True	False	True	False	False	False
1	5	True	False	True	False	False	False
1	4	False	False	False	False	False	False
1	0	False	False	False	False	False	False
1	2	False	False	True	False	False	False
2	4	False	False	False	False	False	False
2	4	False	False	False	False	False	False
2	4	False	False	False	False	False	False
2	12	False	False	False	False	False	False
2	4	False	False	True	False	False	False
2	6	False	False	False	False	False	False
2	4	False	False	False	False	False	False
2	4	False	False	False	False	False	False
2	0	False	False	False	False	False	False
2	10	False	False	False	False	False	False

Figure 3: Časť datasetu po spracovaní

1.1.6 Odstránenie outlierov

Dataset taktiež obsahoval aj outlier hodnoty, ktoré sme sa rozhodli pre zlepšenie kvality trénovania odstrániť.

```

1 dframe = dframe[(dframe['Price'] >= 800) & (dframe['Price'] <=
85000)]
2 dframe = dframe[(dframe['Mileage'] >= 0) & (dframe['Mileage'] <=
500000)]
3 dframe = dframe[(dframe['Engine volume'] >= 0) & (dframe['Engine
volume'] <= 4.5)]
```

Stĺpce, v ktorých sme odhalili outliery boli 'Price', 'Mileage' a 'Engine volume'.

***** Before removing outliers *****						
	Price	Prod. year	Engine volume	Mileage	Airbags	Cylinders
Min values	1	1939	0	0	0	1
Max values	2.63075e+07	2020	20	2.14748e+09	16	16

***** After removing outliers *****						
	Price	Prod. year	Engine volume	Mileage	Airbags	Cylinders
Min values	800	1953	0	0	0	1
Max values	84675	2020	4.5	500000	16	16

Figure 4: Dáta pri spracovaní outlierov

Obrázok č.4 zobrazuje len číselné dáta (nevypisujeme stĺpce čo majú vždy hodnotu True/False [1.0/0.0]).

1.2 Rozdelenie dát na trénovaciu a testovaciu množinu a ich normalizácia

Dáta sme rozdelili na dve množiny a to na vstupnú a výstupnú. Vstupná obsahovala všetky stĺpce, až na stĺpec, ktorý sme chceli aby NS na základe tréningu vedela určiť (Price). Výstupná množina obsahovala práve tento jeden stĺpec. Následne sme hodnoty týchto množín rozdelili do 2 kategórií/množín podľa toho na čo sme dátu chceli použiť. Prvá a najväčšia množina bola trénovacia, táto množina obsahovala 90% dát. Druhá testovacia množina obsahovala 10% dát. Výsledný pomere bol teda (9:1).

V danej podúlohe sme mali za úlohu dátu správne normalizovať. Dáta sme normalizovali podľa postupu povedaného na cvičeniach/seminároch.

♦ Prod. year	Leather interior	♦ Engine volume	♦ Mileage	♦ Cylinders	♦ Doors	♦ Airbags
1953	False	3.20000	100000.00...	4	1	0
2014	False	1.50000	196800.00...	4	2	12
2016	True	2.50000	203073.0...	4	2	4
2009	False	1.90000	196000.00...	4	2	10
2006	False	1.50000	160000.00...	4	2	12
2010	True	2.40000	156000.00...	4	2	12
2012	False	1.80000	288000.0...	4	2	10
2005	False	2.70000	25600.00...	4	2	4

Figure 5: Vybraná časť dát pred normalizáciou

Normalizácia nám ”stlačila” všetky číselné hodnoty do intervalu $<0,1>$ (tak tiež sa nám zmenili aj Boolovské hodnoty zmenili na 1.0/0.0 z True/False).

♦ Prod. year	♦ Leather interior	♦ Engine volume	♦ Mileage	♦ Cylinders	♦ Doors	♦ Airbags
0.00000	0.00000	0.71111	0.20000	0.20000	0.00000	0.00000
0.91045	0.00000	0.33333	0.39360	0.20000	0.50000	0.75000
0.94030	1.00000	0.55556	0.40615	0.20000	0.50000	0.25000
0.83582	0.00000	0.42222	0.39200	0.20000	0.50000	0.62500
0.79104	0.00000	0.33333	0.32000	0.20000	0.50000	0.75000
0.85075	1.00000	0.53333	0.31200	0.20000	0.50000	0.75000
0.88060	0.00000	0.40000	0.57600	0.20000	0.50000	0.62500
0.77612	0.00000	0.60000	0.05120	0.20000	0.50000	0.25000

Figure 6: Vybraná časť dát po normalizácii

1.3 Trénovanie modelov

V tejto kapitole sa venujeme trénovaním a vyhodnotením troch rôznych modelov (DecisionTreeRegressor, RandomForestRegressor, SVM).

1.3.1 Rozhodovací strom

Prvý model, ktorý sme sa rozhodli využiť bol model rozhodovacieho stromu, pre možnosť analýzy sme však museli znížiť jeho "hlbku", to však malo za príčinu horšie natrénovanie modelu (ale zadanie sa nám aj tak podarilo splniť).

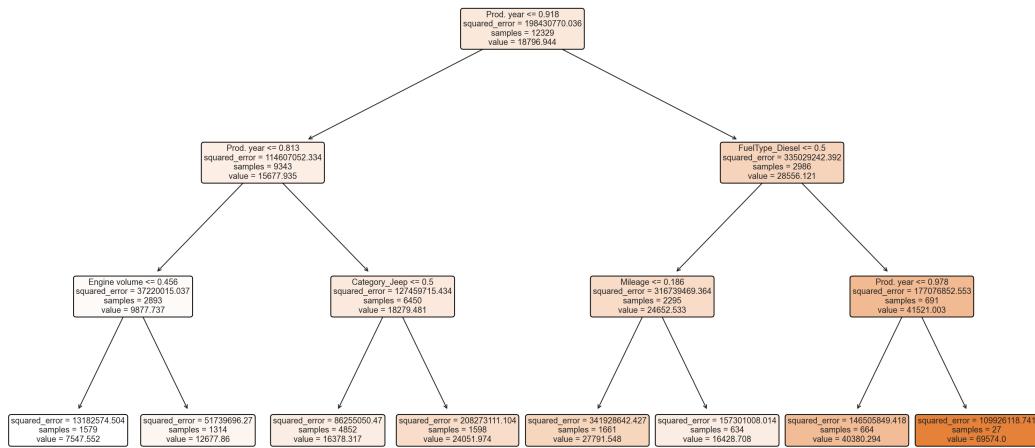


Figure 7: Rozhodovací strom

Graf na obrázku č.7 reprezentuje rozhodovací strom. Graf vieme čítať zhora dole, čiže ako prvú vec pri rozhodovaní sa strom pozrie na hodnotu v stĺpci 'Prod. year', a ak je vyššia ako je zobrazené v danom vrchole tak sa daná vzorka "posunie" doprava a ak je menšia tak doľava. Samples predstavuje počet vzoriek, ktoré sa v danom vrchole nachádzajú a na nich sa teda vykonáva daná podmienka ' \leq '. Value predstavuje hodnotu, ktorú model priradil daným vzorkám v danom vrchole. Squared error predstavuje chybovosť, môžeme si všimnúť, že tá postupne ako ideme stromom dole klesá a to znamená, že model v danom vrchole lepšie aproximuje/predikuje danú hodnotu.

DecisionTreeRegressor (basic dataset):

Metric	Train Set	Test Set
R^2 score	0.343	0.356
MSE	1.30428e+08	1.23976e+08
RMSE	11420.5	11134.4

Figure 8: Výsledky rozhodovacieho stromu

Z obrázku č.8 vyplýva, že sa nám podarilo natrénovať model dostatočne (dosiahli sme kladné R2 skóre).

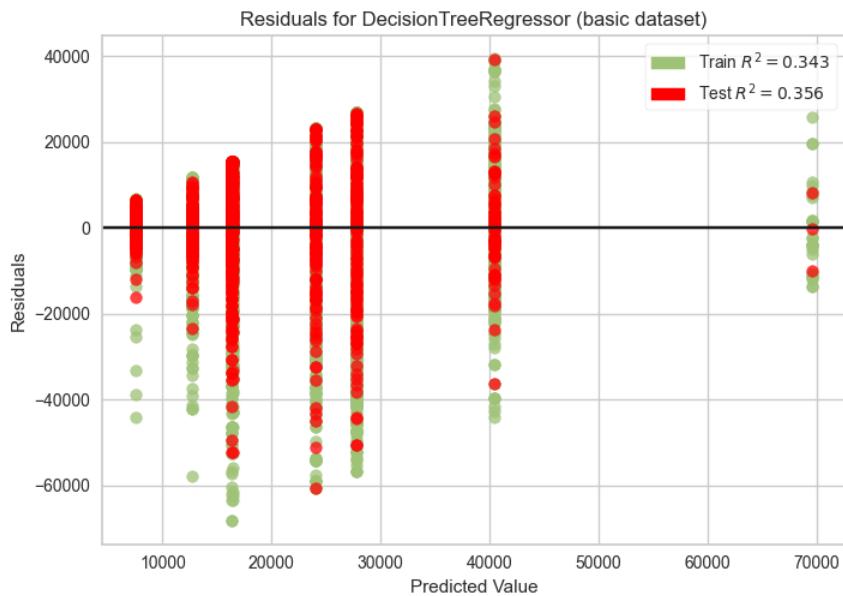


Figure 9: Residual graf pre rozhodovací strom

Obrázok č.9 reprezentuje graf reziduálov modelu rozhodovacieho stromu. Z grafu reziduálov vieme zistiť trend chybovosti nášho modelu, to znamená že v našom konkrétnom prípade náš model bol presnejší pri hádaní nižších hodnôt a mal rastúcu tendenciu sa mýliť s vyššími hodnotami. Vzdialenosť bodu,

kt. reprezentuje reziduál od hodnoty 0 predstavuje o kolko sa model pomýlil či už v kladnom, alebo zápornom smere (či reálna hodnota vozidla bola o danú hodnotu vyššia, alebo nižšia). Zelené body reprezentujú reziduály trénovacích dát a červené body testovacích dát. Vo všeobecnosti platí, že kladné reziduály predstavujú hodnotu o kolko väčšia je reálna hodnota oproti "predikovanej" a naopak. Avšak v tejto dokumentácii sú hodnoty obrátené, je to kvôli tomu, že využitá knižnica vytvára tieto reziduály "opačne". To znamená, že záporné reziduály predstavujú vzťah kedy je reálna hodnota väčšia ako predikovaná!

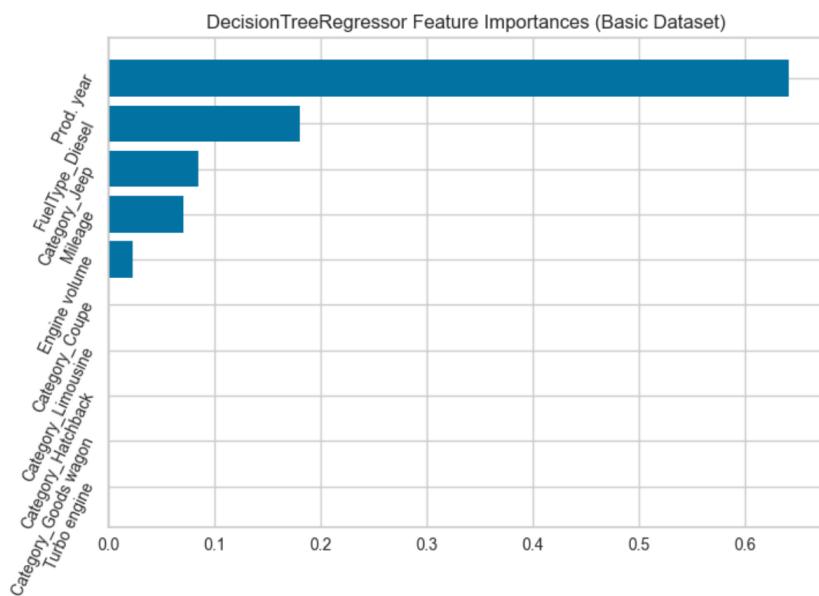


Figure 10: Najdôležitejšie príznaky pre rozhodovací strom

Z obrázku č.10 vieme zistiť, že rozhodovací strom za najdôležitejšie príznaky považoval 'Prod. year' a 'FuelType_Diesel'.

1.3.2 Ensemble model (Random Forest)

Druhý model, na ktorý sme sa zamerali bol RandomForestRegressor, zároveň sme mu podobne ako ostatným modelom zobrazili najdôležitejšie príznaky a zhodnotili jeho trénovanie.

RandomForestRegressor (basic dataset):

Metric	Train Set	Test Set
R^2 score	0.711	0.677
MSE	5.74444e+07	6.22092e+07
RMSE	7579.21	7887.28

Figure 11: Výsledky RandomForest modelu

Z obrázku č.11 vyplýva, že sa nám podarilo natrénovať model, tak aby splňal aj podmienku pre minimálne R2 skóre zo zadania (dosiahli sme R2 skóre vyššie ako 0.5).

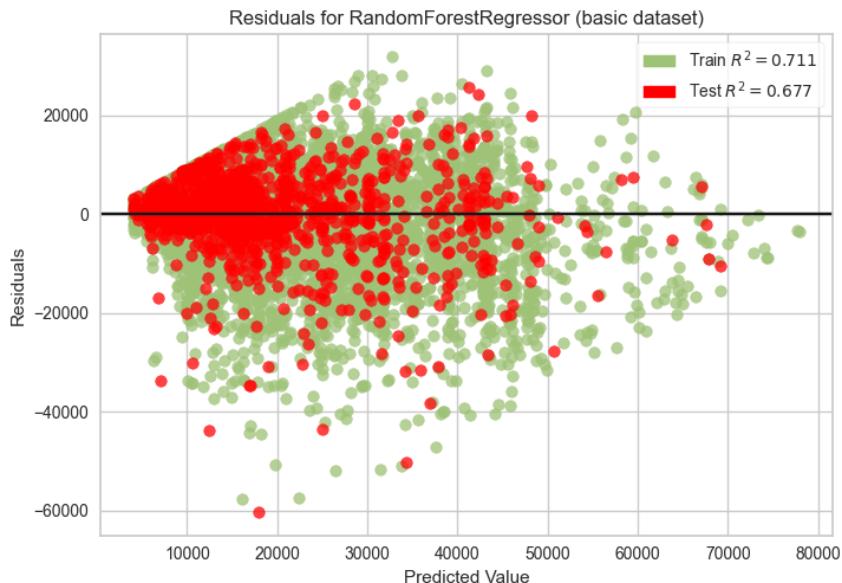


Figure 12: Residual graf pre RandomForest model

Obrázok č.12 reprezentuje graf reziduálov modelu. Záporné reziduály predstavujú vzťah kedy je reálna hodnota väčšia ako predikovaná a kladné reziduály predstavujú vzťah kedy je predikovaná hodnota väčšia ako reálna hodnota.

Na obrázku vidíme, že oproti predošlému modelu majú reziduály (aj kladné aj záporné) nižsie maximálne (absolútne) hodnoty.

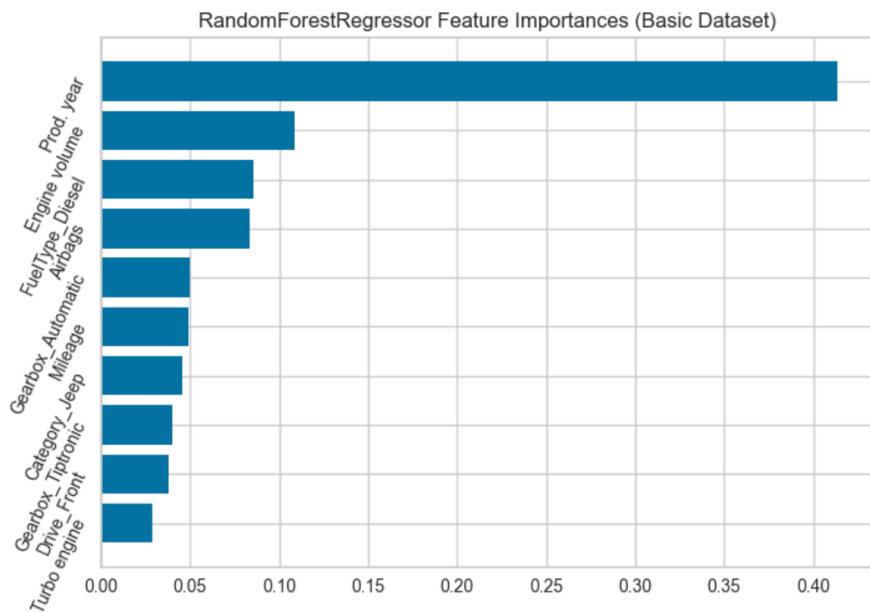


Figure 13: Najdôležitejšie príznaky pre RandomForest model

Ďalšiou úlohou bolo vytvoriť graf pre najdôležitejšie príznaky daného modelu. Po vytvorení grafu sme ho zanalyzovali a zistili, že, že náš model považoval za najdôležitejší príznak s veľkým prehľadom 'Prod. year' (pre lepšiu čitateľnosť grafu sme vybrali do grafu len TOP10 najdôležitejších príznakov).

1.4 Model SVM

Ako tretí model sme využili model SVM, v knižnici sklearn sme vybrali regressor typ modelu SVM (kedže nechceme kategorizovať, ale určovať hodnoty), preto sa v nasledujúcich obrázkoch bude spomínať SVR, ide však stále o model SVM.

SVR (basic dataset):

Metric	Train Set	Test Set
R^2 score	0.625	0.608
MSE	7.43596e+07	7.5539e+07
RMSE	8623.2	8691.32

Figure 14: Výsledky SVM modelu

Z obrázku č.14 vyplýva, že sa nám podarilo natrénovať model, tak aby splňal aj podmienku pre minimálne R2 skóre zo zadania (dosiahli sme R2 skóre vyššie ako 0.5).

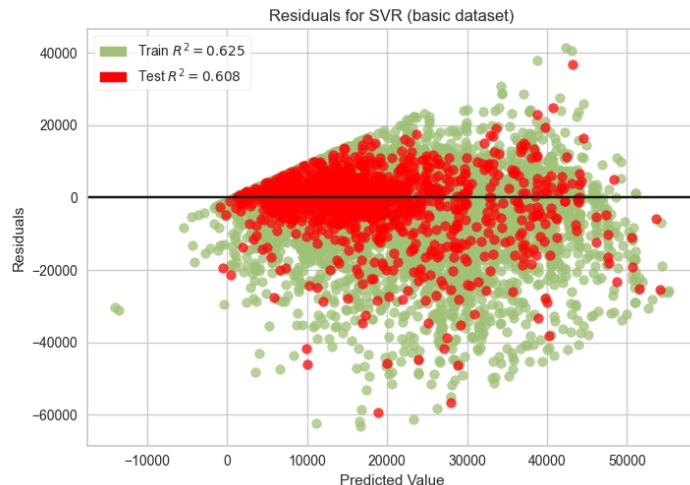


Figure 15: Residual graf pre SVM model

Obrázok č.15 reperzentuje graf reziduálov modelu. Zaujímavostou na tomto obrázku, je že model pri niektorých vzorkách predikoval dokonca zápornú cenu vozidla.

1.4.1 Porovnanie modelov

V tejto časti si porovnáme tri natréновané modely a určíme si najúspešnejší model, ktorý využijeme ešte neskôr v tomto zadaní pre jeho trénovanie na podmnožinách príznakov datasetu. A taktiež vieme skonštatovať, že mal väčšie maximálne hodnoty reziduálov (v absolútnych hodnotách) ako predošlý model.

Model	Nastavenia modelu	[R2 skóre, RMSE] na trénovacích dátach	[R2 skóre, RMSE] na testovacích dátach
DecisionTreeRegressor	(max_depth=3, random_state=71)	[0.343, 11420.5]	[0.356, 1134.4]
RandomForestRegressor	(n_estimators=300, max_depth=7, random_state=71)	[0.711, 7579.2]	[0.677, 7887.3]
SVM (SVR)	(kernel='rbf', C=9500, gamma=0.7)	[0.625, 8623.2]	[0.608, 8691.3]

Table 1: Porovnanie modelov

Z tabuľky č.1 vyplýva, že najlepšie natrénovaný model bol RandomForestRegressor. Podobné zistenie sme zanalyzovali už počas vysvetľovania reziduálnych grafov, kde sme si všimli, že práve tento model bol zo všetkých troch najpresnejší. Pri všetkých modeloch si však môžeme všimnúť, že modely mají najväčší problém s určením ceny "stredne drahých" áut (na grafoch vidíme, že trend chybovosti aj kladných aj záporných reziduálov sa zväčšoval s narastajúcou cenou, ale v istom momente začal klesať).

2 Druhá časť

V tejto časti zadania sme sa venovali redukcii dimenzie pomocou príznakov a pomocou PCA.

2.1 Redukcia dimenzie pomocou 3 príznakov

Tri príznaky, ktoré sme sa rozhodli využiť pri redukcii boli 'Prod. year', 'Mileage' a 'Engine Volume'.

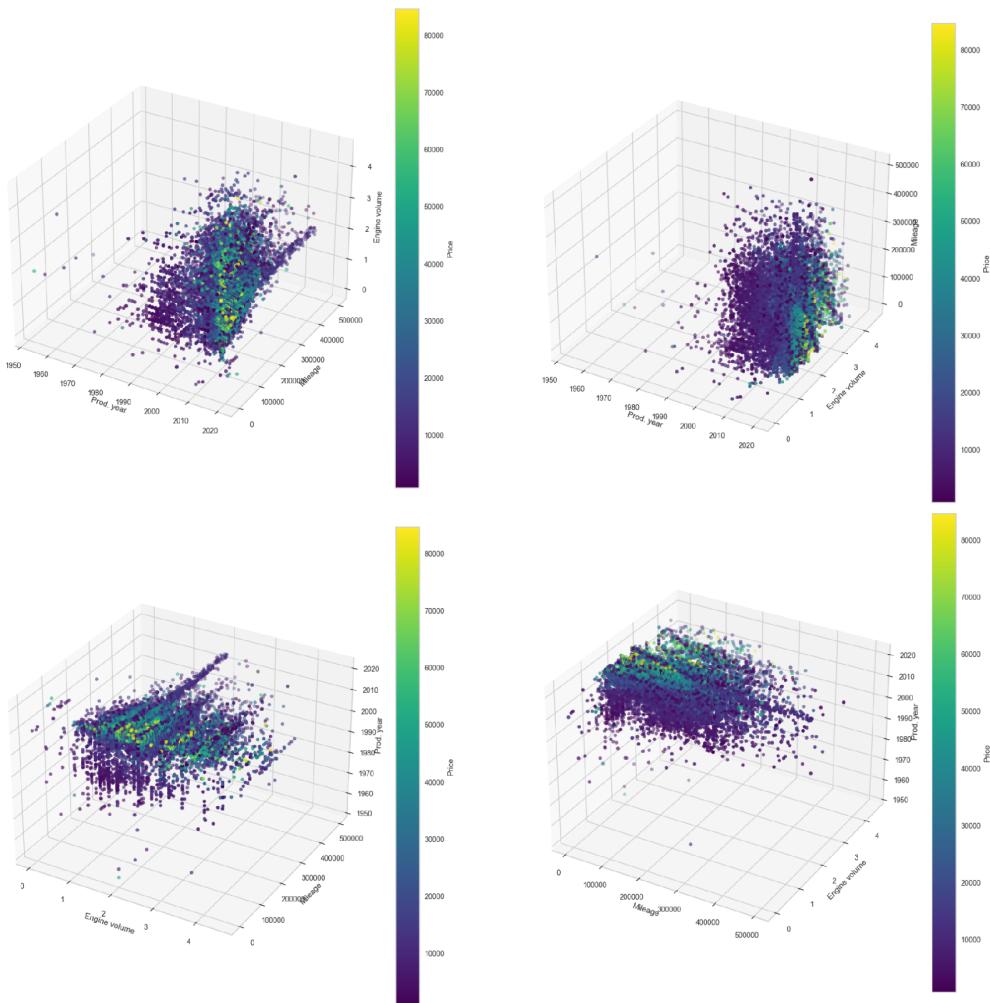


Figure 16: 3D Point Graf troch príznakov

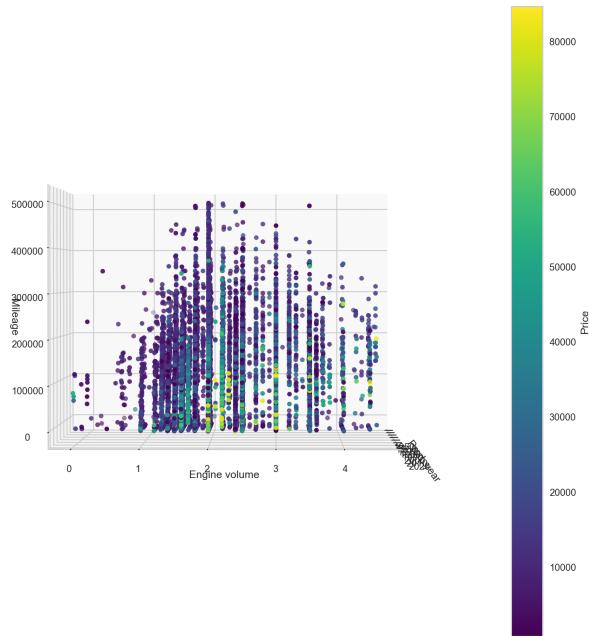


Figure 17: 2D pohľad na lepšiu vizualizáciu 'Engine Volume'

Obrázok č.16 zobrazuje 3D point graf spomínaných príznakov. Farba každého bodu reprezentuje cenu vozidla (čím je farba svetlejšia, tým je cena vozidla vyššia), vďaka týmto vlastnostiam môžeme zanalyzovať dané dátu. Z grafu teda vieme vyčítať, že čím je vozidlo novšie a zároveň má čo najmenej nazadené tým viac stúpa jeho cena, čo sa týka objemu motora, tak tam sa skupujú najdrahšie vozidlá od honoty 2l a vyššie. Zjednodušene povedané so stúpajúcou hodnotou 'Mileage' klesá cena, so stúpajúcou hodnotou 'Prod. year' stúpa taktiež aj cena, a objem motora s hodnotou 2l a vyššie je podľa grafu indikátor vyššej ceny (v okolí [0-1] sa nachádzaju iba lacnejšie autá, pre lepšiu vizualizáciu sme zhotovali aj obrázok č.17).

2.2 Redukcia dimenzie pomocou PCA

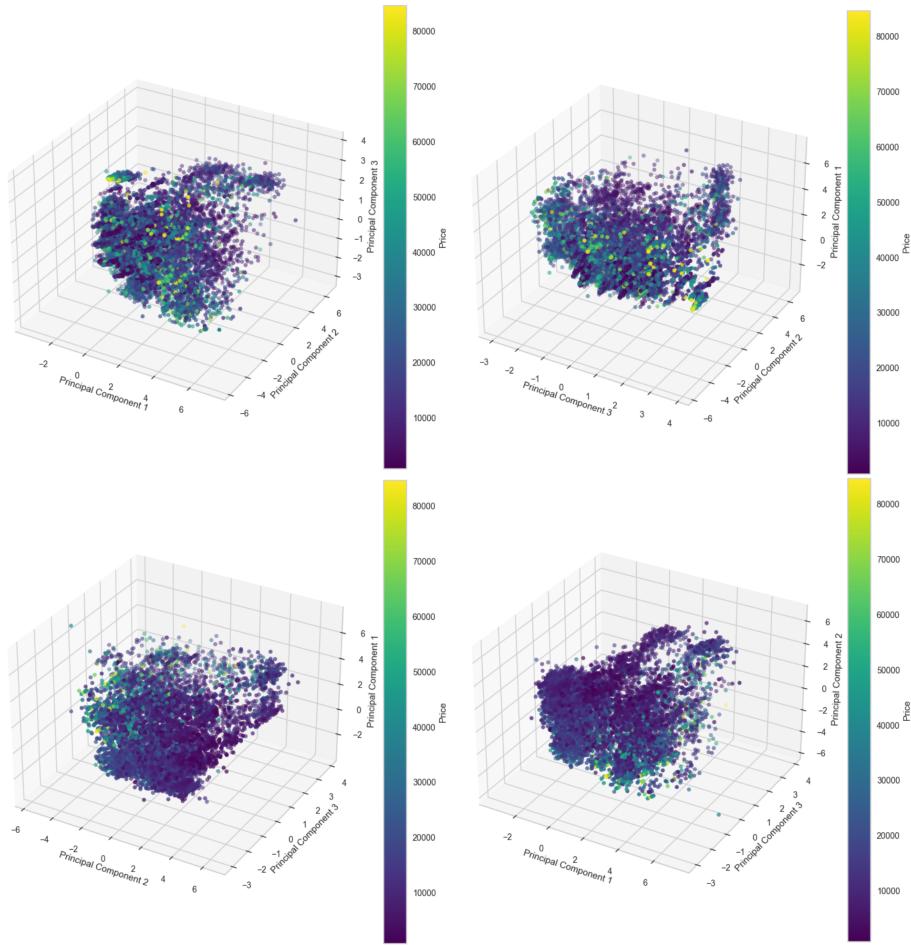


Figure 18: 3D Point Graf minimalizovanej množine pomocou PCA

Po redukcii dimenzie pomocou využitia PCA sme dostali tri komponenty, ktoré sú zobrazené aj na obrázku č.18. Môžeme si všimnúť, že cenu aút najviac ovplyvňuje hlavný komponent 1 a následne hlavný komponent 2 (všimnúť si to vieme najmä keď sa zameriame na spodné grafy, kde vidíme, že pokial Principal Component 1 nemá aspoň hodnotu 0 tak autá sú skoro všetky lacné). Následne si môžeme všimnúť, že so snižujúcimi sa hodnotami druhého komponentu vzrástá cena, najmenej ovplyvňujúci komponent je Principal Component 3.

3 Tretia časť

V tretej časti sme sa zamerali na podmnožiny príznakov, ktoré sme využili na natrénovanie nášho najúspešnejšieho modelu z prvej časti (Random Forest Regressor).

3.1 Trénovanie na podmnožine príznakov - Korelačná matica

Pri hľadaní vhodných kandidátov sme sa sústredili na koreláciu ceny s ostatnými dátami (hľadali sme čo najväčšiu absolútну hodnotu v stĺpci/riadku Price). Na základe tejto analýzy sme vybrali nasledujúce dátá 'Prod. year', 'Category_Jeep', 'Leather interior', 'FuelType_Diesel', 'Mileage' (so stúpajúcou 'Mileage' klesá cena, preto hodnota korelácie v matici bola záporná, ostatné vybrané dátá stúpajú zároveň s cenou [True=1.0, False=0.0], preto ich hodnota v matici bola kladná).

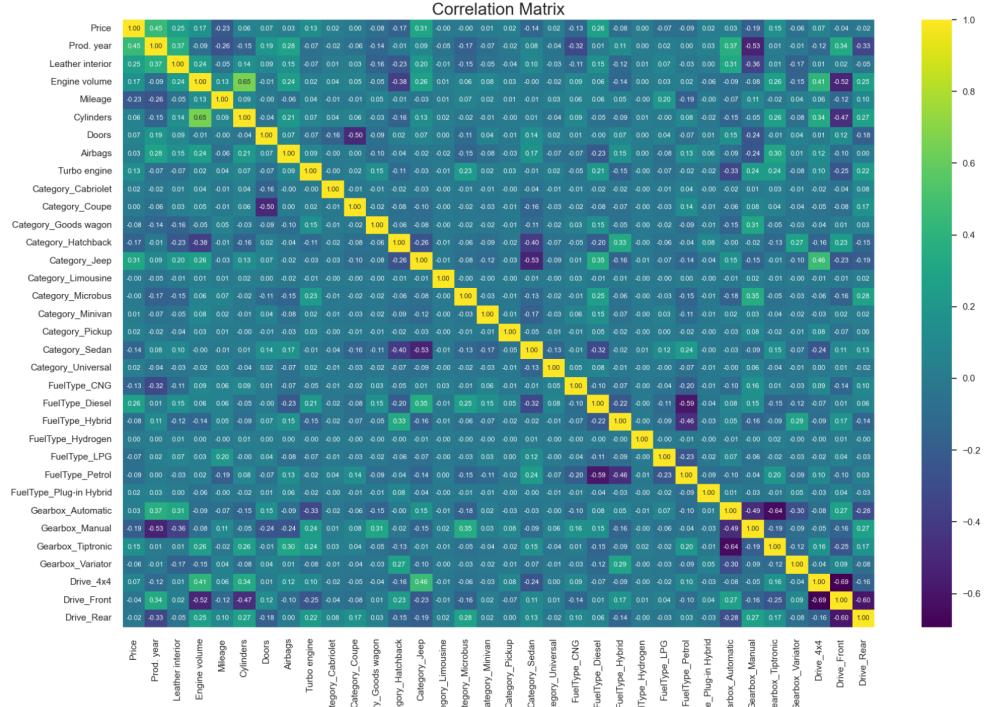


Figure 19: Korelačná matica datasetu

RandomForestRegressor (correlation matrix):

Metric	Train Set	Test Set
R^2 score	0.473	0.454
MSE	1.04658e+08	1.05083e+08
RMSE	10230.3	10251

Figure 20: Výsledky RandomForest modelu

Z obrázku č.20 vyplýva, že sa model s danou podmnožinou príznakov natrénoval horšie ako model s celým datasetom, bližšie sa k daným výsledok budeme venovať v neskoršej kapitole.

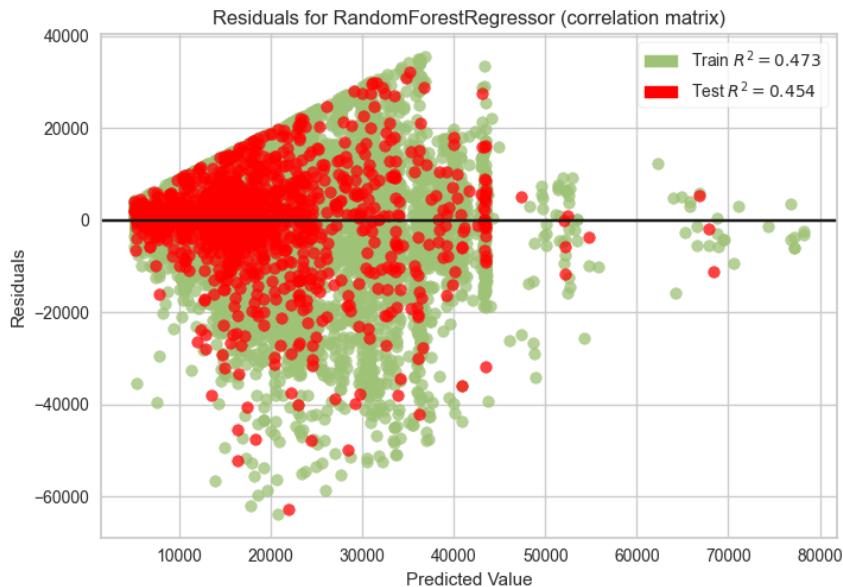


Figure 21: Residual graf pre RandomForest model

Obrázok č.21 reprezentuje graf rezidúalov modelu. Pri tomto grafe si môžeme všimnúť že model predikoval väčšinu hodnôt v intervale $\langle 800, 40000 \rangle$.

3.2 Trénovanie na podmnožine príznakov - Dôležitosť príznakov z ensemble modelu

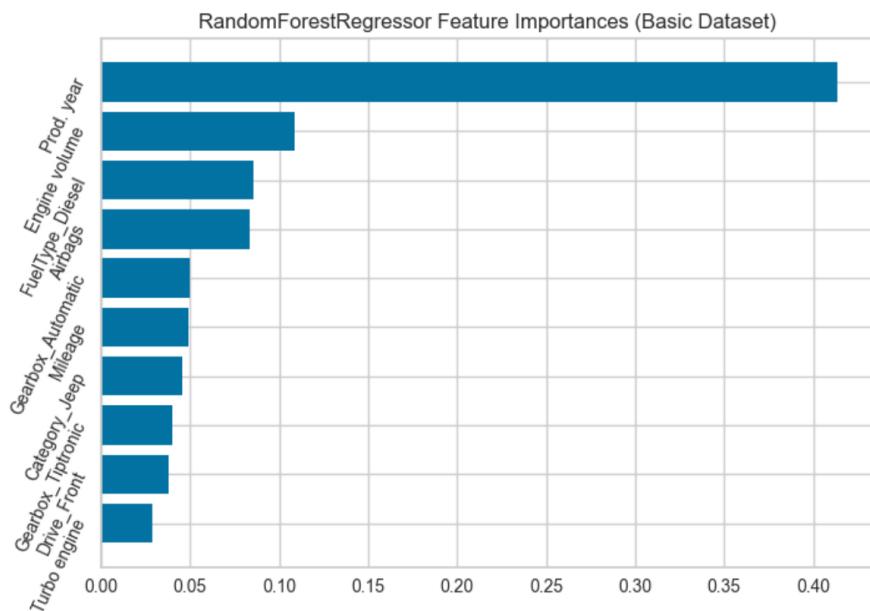


Figure 22: Najdôležitejšie príznaky pre RandomForest model

V tejto časti sme vyberali príznaky na základe ich dôležitosti podľa ensemble modelu. Tieto príznaky sme už zistili v predošlej časti zadania a budeme sa teda na ne odkazovať (obrázok č.22). Podobne ako pri korelačnej matici sme vybrali 5 najdôležitejších príznakov, v tomto prípade išlo o 'Prod. year', 'Engine volume', 'FuelType_Diesel', 'Airbags', 'Gearbox_Automatic'.

RandomForestRegressor (top features):

Metric	Train Set	Test Set
R^2 score	0.634	0.609
MSE	7.25879e+07	7.52578e+07
RMSE	8519.86	8675.13

Figure 23: Výsledky RandomForest modelu

Z obrázku č.23 vyplýva, že sa model s danou podmnožinou príznakov natrénoval horšie ako model s celým datasetom, bližšie sa k daným výsledkom budeme venovať v neskoršej kapitole.

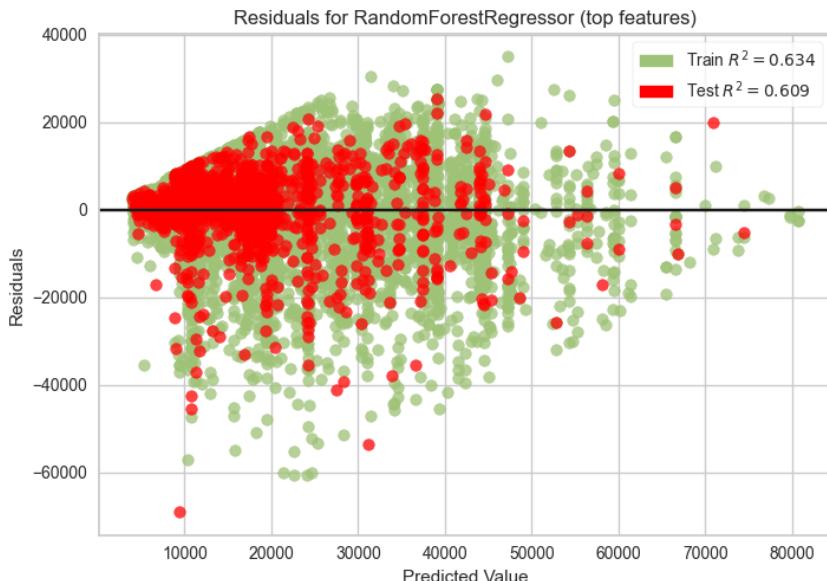


Figure 24: Residual graf pre RandomForest model

Obrázok č.24 reprezentuje graf reziduálov modelu. Pri tomto grafe už vidíme zlepšenie oproti poslednému pokusu, model odhaduje hodnoty rovnomernejšie po celej x-ovej osi.

3.3 Trénovanie na podmnožine príznakov - Variancia pomocou PCA

RandomForestRegressor (PCA):

Metric	Train Set	Test Set
R^2 score	0.497	0.456
MSE	9.97126e+07	1.04702e+08
RMSE	9985.62	10232.4

Figure 25: Výsledky RandomForest modelu

Z obrázku č.25 vyplýva, že sa model s danou podmnožinou príznakov natrénoval horšie ako model s celým datasetom, bližšie sa k daným výsledkom budeme venovať v nasledujúcej kapitole.

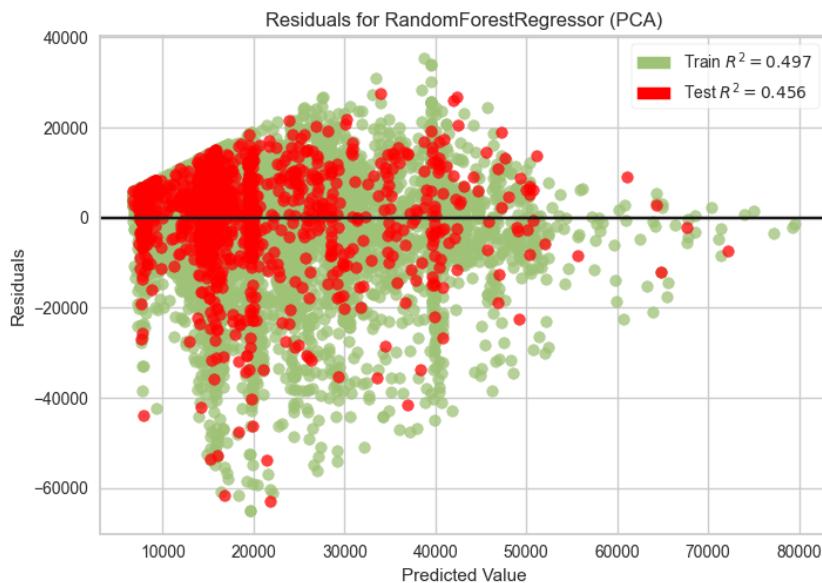


Figure 26: Residual graf pre RandomForest model

Obrázok č.26 reprezentuje graf reziduálov modelu. Pri tomto grafe už vidíme zlepšenie oproti pokusu s podmnožinou vytvorenou na základe korelačnej matice, model odhaduje hodnoty rovnomernejšie po celej x-ovej osi podobne ako pri predošлом pokuse. Avšak môžeme si všimnúť, hlavne pri záporných reziduáloch, že väčšina predikcií je do hodnoty 50000.

3.4 Porovnanie výsledkov trénovaní

Dataset "mód"	[R2 skóre, RMSE] na trénovacích dátach	[R2 skóre, RMSE] na testovacích dátach
Basic	[0.711, 7579.2]	[0.677, 7887.28]
Correlation Matrix	[0.473, 10230.3]	[0.454, 10251.0]
Top Features	[0.634, 8519.9]	[0.609, 8675.1]
PCA	[0.497, 9985.6]	[0.456, 10232.4]

Table 2: Porovnanie modelov na základe dát

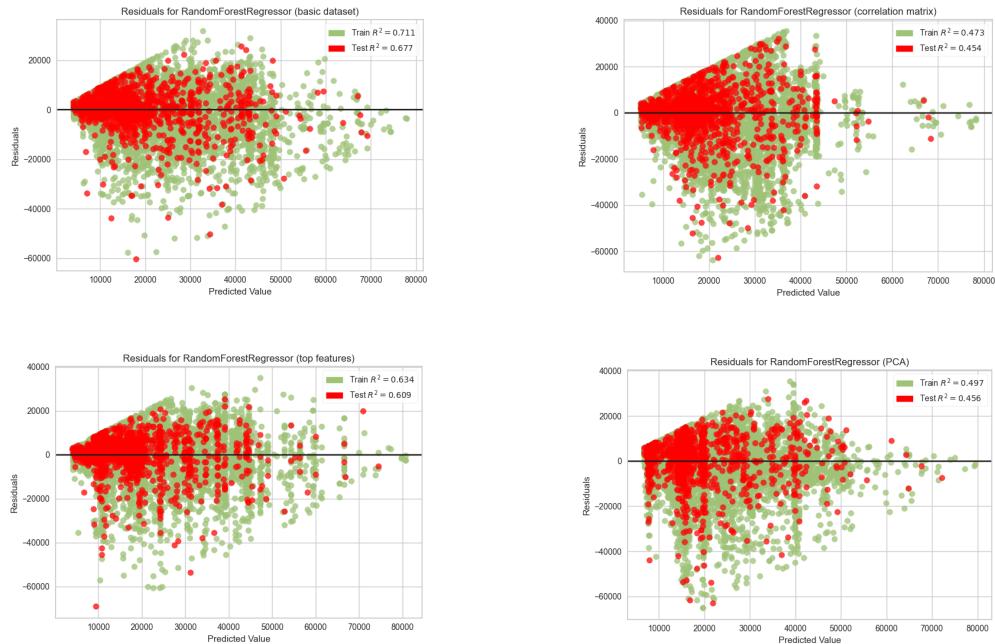


Figure 27: Residual grafy pre RandomForest model

Na základe tabuľky č.2 môžeme vyčítať, že model bol najúspešnejší, keď mal k dispozícii celý dataset, druhý najlepší bol pri využití top 5 najdôležitejších

príznakov, následne bol model s PCA datasetom a najhoršie skončil model trénovaný cez príznaky získané v korelačnej matici. Taktiež si môžeme z grafov všimnúť, že pri dvoch najúspešnejších pokusoch model častejšie predikoval aj vyššie ceny áut.