SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

AIS ID: 92320

STROMY, STROJE, HLASOVANIA A REDUKCIA DIMENZIE

ZADANIE

Predmet:	I-SUNS – Strojové učenie a neurónové siete
Prednášajúci:	prof. Dr. Ing. Miloš Oravec
Cvičiaci:	Ing. Marián Šebeňa

Bratislava 2024 **Bc. Lukáš Patrnčiak**

Obsah

Zoznam obrázkov a tabuliek

Zoznam skratiek

EDA	Exploratory Data Analysis
MSE	Mean Square Error
PCA	Principal Component Analysis
R2	Koeficient determinácie
ReLU	Rectified Linear Unit
SVM	Support Vector Machine

Úvod

Cieľom tohto zadania je v programovacom jazyky Python implementovať program, ktorý bude predpovedať cenu letenky na základe poskytnutých dát z AIS.

Najprv bude potrebné načítať dáta a pripraviť ich na spracovanie modelmi strojového učenia - odstrániť prázdne hodnoty, duplikáty, identifikátory a pre textové stĺpce bolo potrebné použiť chodné kódovanie. Následne budú dáta rozdelené na trénovaciu a testovaciu množinu a následne na vstupnú (ktorá bola neskôr normalizovaná) a výstupnú množinu.

Potom prebehne trénovanie rozhodovacieho stromu, vybraného stromového súboru - ensemble modelu a SVM. Tieto modely búdu vyhodnotené na trénovacej a testovacej množine pomocou MSE a R2 a výsledky budú vizualizované tak, aby bolo možné analyzovať reziduály.

Následne bude pozorované, čo s dátami robí redukcia dimenzie a na záver prebehne výber podmnožiny príznakov - bol vybraný najúspešnejší model z prvej časti zadania a bol opäť natrénovaný pre zmenšenú množinu.

Na záver práce prebehne výber 5 príznakov na základe korelačnej matice a ich analýza pomocou EDA a následne bude natrénovaná neurónová sieť.

1 Načítanie a príprava dát

V tejto kapitole bude popísaný spôsob načítania a prípavy dát, pre ich neskoršie použitie na spracovanie vybranými modelmi strojového učenia. Spracovávané dáta sú v súbore *flight_data.csv*. Pre spracovanie, trénovanie a vykreslenie údajov boli použité knižnice:

- NumPy[1]: Pre matematické výpočty.
- Pandas[2]: Manipulácia s tabuľkovými dátami a ich analýza.
- Seaborn[3]: Vysokoúrovňová vizualizácia dát so zameraním na štatistiku.
- Matplotlib.pyplot[4]: Základná knižnica na tvorbu grafov s nízkoúrovňovou kontrolou nad vizualizáciami.
- skLearn[5]: Knižnica určená pre klasické metódy strojového učenia. Obsahuje množstvo algoritmov pre klasifikáciu, regresiu,...
- category_encoders[6] - Knižnica obsahujúca kódovania údajov (použitá na Target Encoder)
- Tensorflow (Keras)[7]: Tvorba a trénovanie modelov strojového učenia a neurónových sietí. API Keras zjednodušuje tvorbu a trénovanie neurónových sietí.

Po načítaní dát z daného súboru nasledoval ich výpis, aby sme zistili, rôzne informácie o nich, predovšetkým koľko duplicít a chýbajúcich vzoriek obsahujú a aký je ich celkový počet. Najprv boli rozdelené množiny na textové a číselné, pričom ak sa vyskytol v číselnej množine text, ten bol nahradený mediánom zvyšných číselných hodnôt a potom boli odstránené stĺpce, ktoré obsahujú identifkátory ('ID' a 'flight'). Potom boli odstránené outliery (neobvyklé hodnoty, odstránenie zabezpečené funkciou *outliers()*), prázdne hodnoty (NaN) a duplicity.

[1] https://numpy.org/
[2] https://pandas.pydata.org/
[3] https://seaborn.pydata.org/
[4] https://matplotlib.org/stable/api/pyplot_summary.html
[5] https://scikit-learn.org/stable/
[6] https://contrib.scikit-learn.org/category_encoders/
[7] https://www.tensorflow.org/guide/keras

Upresnenie jednotlivých stĺpcov (množín) v dátovej množine:

- *ID* - identifikátor zoznamu.
- *Airline* - názov leteckej spoločnosti
- *Flight* - identifikátor letu
- *Source city* - mesto, z ktorého lietadlo odlieta
- *Departure time* - časový rámec odletu lietadla
- *Stops* - počet zastávok letu
- *Arrival time* - časový rámec príletu lietadla
- *Destination city* - mesto, kde lietadlo pristane.
- *Class* - trieda cestovania
- *Duration* - dĺžka letu
- *Days left* - počet dní do odletu
- *Price* - cena letenky

Pre zakódovanie textových množín/stĺpcov boli použité nasledovné kódovania:

- *LabelEncoding* - používa sa zvyčajne pre kategorické premenné, ktoré sú usporiadané. Tento typ kódovania kóduje príslušné hodnoty ako celé čísla a bol použitý pre množiny *class, stops, departure_time* a *arrival_time*.
- *Target Encoding* - používa sa zvyčajne v prípadoch, keď je počet kategórií veľký a chceme zachovať vzťah medzi kategóriami a cieľovou premennou bez nadmerného nárastu počtu stĺpcov. Pre každú kategóriu v stĺpci sa vypočíta priemerná hodnota cieľovej premennej (*price*). Kategória sa nahradí touto priemernou hodnotou. Takto zakódované množiny sú: *airline, source_city* a *destination_city*.

Dáta boli následne rozdelené na trénovaciu a testovaciu množinu a následne na vstupnú (X) a výstupnú (Y) množinu. Vstupná množina, ktorá bola škálovaná *SdandardScaler*-om (nazývaná tiež z-score normalizácia, kde sa hodnoty škálujú na normálne rozdelenie so strednou hodnotou 0 a smerodajnou odchýlkou 1) a obsahuje všetky množiny dát, okrem množiny *price*, ktorá sa nachádza vo výstupnej množine Y.

2 Trénovanie modelov

V tejto kategórii budú popísané trénovania troch vybraných modelov a ich vyhodnotenia na trénovacej a testovacej množine pomocou MSE, resp. RMSE a R2, vrátane vizualizácie výsledkov tak, aby bolo možné analyzovať ich reziduály. Trénovaine modelov zabezpečuje funkcia *train_model()* a ich vyhodnotenie (vrátane vizualizácií) funkcia *evaluate_model()*.

- **(R)MSE**: Meria priemernú veľkosť chýb medzi predikovanými a skutočnými hodnotami. Čím nižšia hodnota RMSE, tým lepšie model "sedí"na dáta. To znamená, že predikované hodnoty sú bližšie k skutočným hodnotám. RMSE má rovnakú jednotku ako pôvodné dáta, čo uľahčuje interpretáciu (na rozdiel od MSE).

- **R2**: Vyjadruje, aké množstvo variability závislej premennej (to, čo chceme predikovať) je vysvetlené nezávislými premennými (vstupnými údajmi) v modeli. Vhodná hodnota R2 sa pohybuje od 0 do 1. Čím bližšie je hodnota k 1, tým väčšia časť variability je vysvetlená modelom. Hodnota 0 znamená, že model nevysvetľuje žiadnu variabilitu. R2 je bezrozmerná hodnota.

Rozptýlenie reziduálov: *Reziduál je rozdiel medzi skutočnou hodnotou a hodnotou predikovanou modelom. Inými slovami, predstavuje chybu, ktorej sa model dopustil pri predpovedaní.* Ak sú reziduá rovnomerne rozptýlené okolo horizontálnej osi (y = 0), bez zjavného trendu alebo vzoru, znamená to, že model nemá systematické chyby (chyby modelu sú náhodné a neexistuje žiadna systematická odchýlka) a pravdepodobne je správne nastavený . Ide o indikátor dobrého modelu.

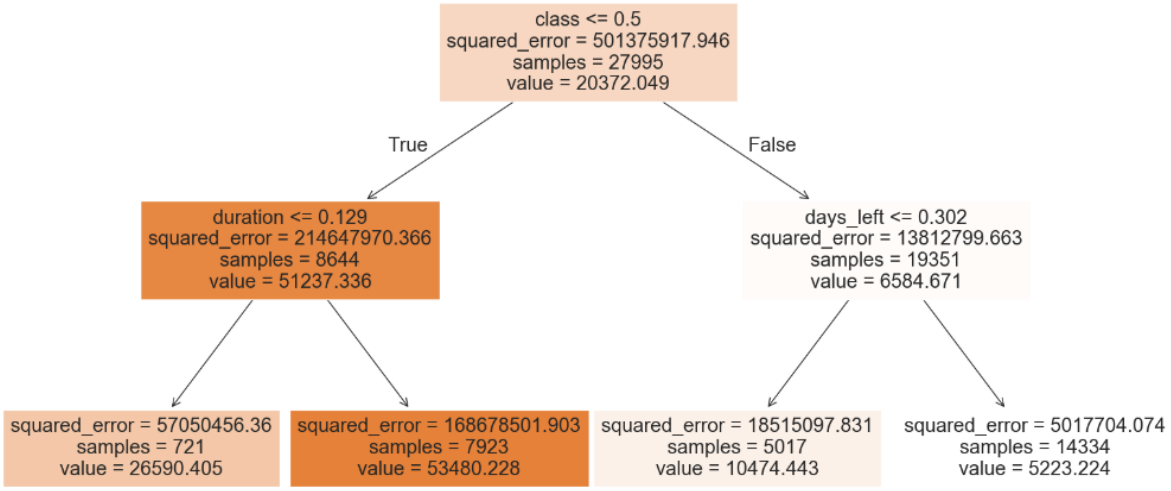
Ak reziduá ukazujú na konkrétny vzor, napríklad sa zhlukujú okolo určitého bodu, znamená to, že model nefunguje optimálne a môže byť potrebné zvážiť iný prístup, transformáciu dát alebo zlepšiť model. Veľkosť reziduálov nám hovorí o veľkosti chýb modelu. Čím sú reziduály menšie, tým presnejšie model predpovedá.

2.1 Rozhodovací strom

Rozhodovací strom je model, ktorý predstavuje rozhodovací proces ako stromovú štruktúru. Každý uzol v strome reprezentuje atribút (príznak) a každá vetva predstavuje možnú hodnotu tohto atribútu. Listy stromu reprezentujú triedy alebo hodnoty, ktoré chceme predpovedať.

Aby bol rozhodovací strom korektne vykreslený (t.j. nie príliš veľký), boli upravené parametre stromu:

- *Maximálna hĺbka* - Tento parameter obmedzuje maximálnu hĺbku stromu. V tomto prípade môže mať strom najviac 2 úrovne (od koreňa po najvzdialenejšie listy)
- *Minimálny počet vzoriek* - Tento parameter určuje minimálny počet vzoriek, ktoré musí uzol obsahovať, aby sa mohol ďalej rozvetviť. V tomto prípade, ak uzol obsahuje menej než 6 vzoriek, už sa nebude ďalej rozvetvovať a stane sa listom.

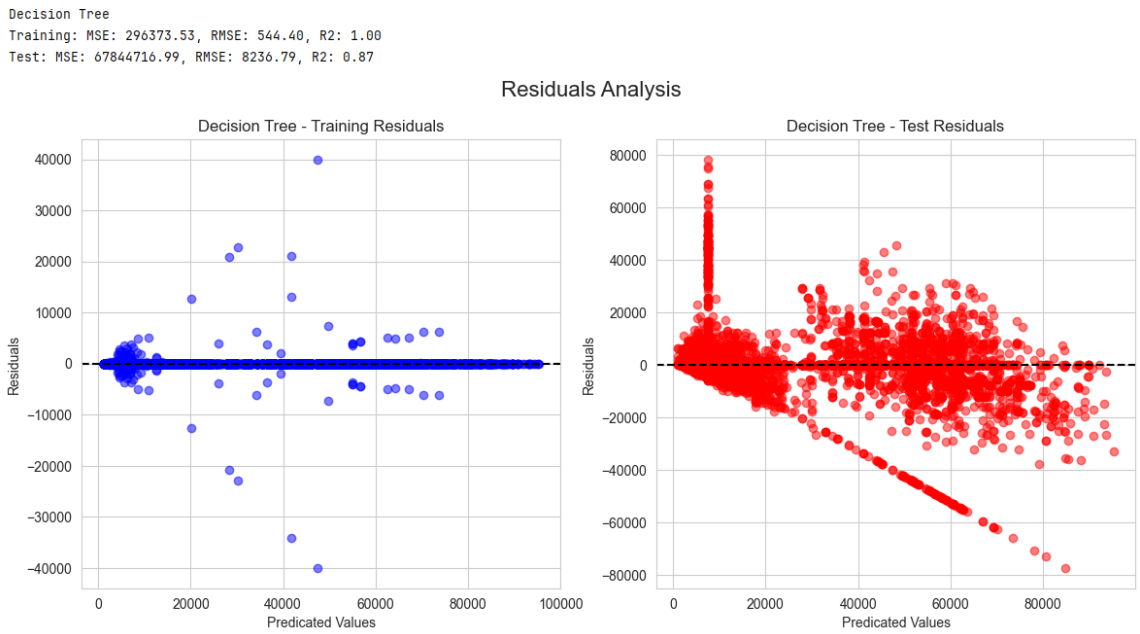


Obrázok 1: Rozhodovací strom - vizualizácia

Strom začína delením na základe množiny class <= 0.5, čo rozdeľuje dáta do dvoch skupín. Ľavá vetva ďalej delí na základe množiny duration, zatiaľ čo pravá vetva delí na základe množiny days_left. Na každom uzle sa znižuje hodnota štvorcovej chyby (squared error), čo naznačuje, že jednotlivé rozdelenia zlepšujú presnosť predikcie. Hodnoty **samples** a **value** ukazujú, koľko vzoriek a aká je priemerná hodnota cieľovej premennej v každom uzle.

Je potrebné poznamenať, že tento uvedený strom bol znázornený s uvedenými parametrami za účelom zobrazenia a popísania v zadaní - vyhodnotenie rezíduii preto nekorešponduje s týmto stromom.

Na obrázku nižšie môžeme vidieť vizualizáciu reziduí a taktiež vypísania (R)MSE a R2 skóre pre trénovaciu a testovaciu množinu. Ako základ pre porovnanie modelov budeme brať R(MSE) a R2 skóre pre testovaciu množinu.



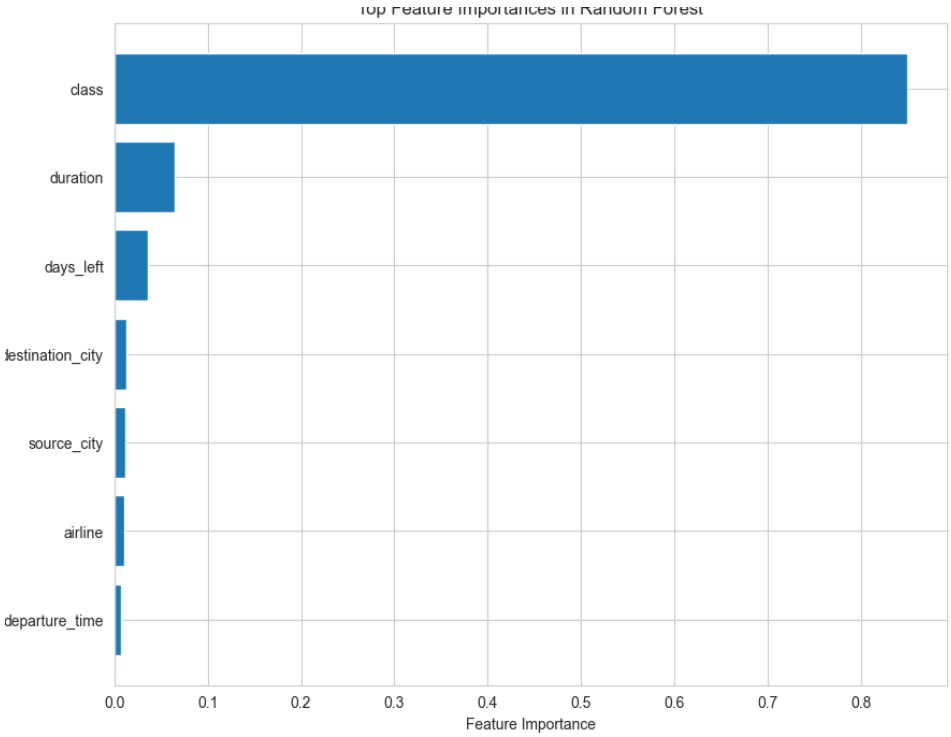
Obrázok 2: Rezíduá a rozhodovací strom s príslušnými (R)MSE a R2 hodnotami

2.2 Stromový súborový model

Random Forest je jedným z najpopulárnejších algoritmov pre ensemble learning, ktorý patrí medzi metódy zlepšovania výkonu modelu kombinovaním viacerých jednoduchých modelov. Tento prístup sa nazýva bagging (Bootstrap Aggregating). Random Forest je konkrétne ensemble model tvorený množstvom rozhodovacích stromov, ktoré pracujú spoločne a vytvárajú silný model.

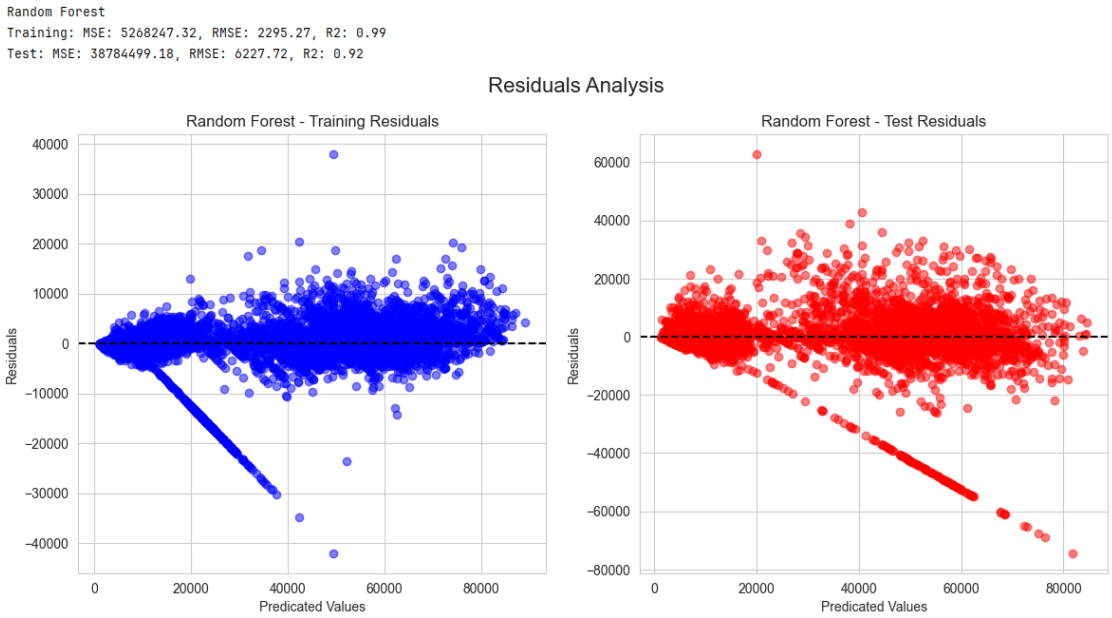
V tomto prípade bolo pri vyhodnotení vytvorených 100 rozhodovacích stromov.

Obrázok nižšie zobrazuje dôležitosť vstupných parametrov, pričom tento počet bol zredukovaný na 7 najdôležitejších.



Obrázok 3: Stromový súbor - Random Forest

Na obrázku nižšie môžeme vidieť vizualizáciu reziduí a taktiež vypísania (R)MSE a R2 skóre pre trénovaciu a testovaciu množinu. Ako základ pre porovnanie modelov budeme brať R(MSE) a R2 skóre pre testovaciu množinu.



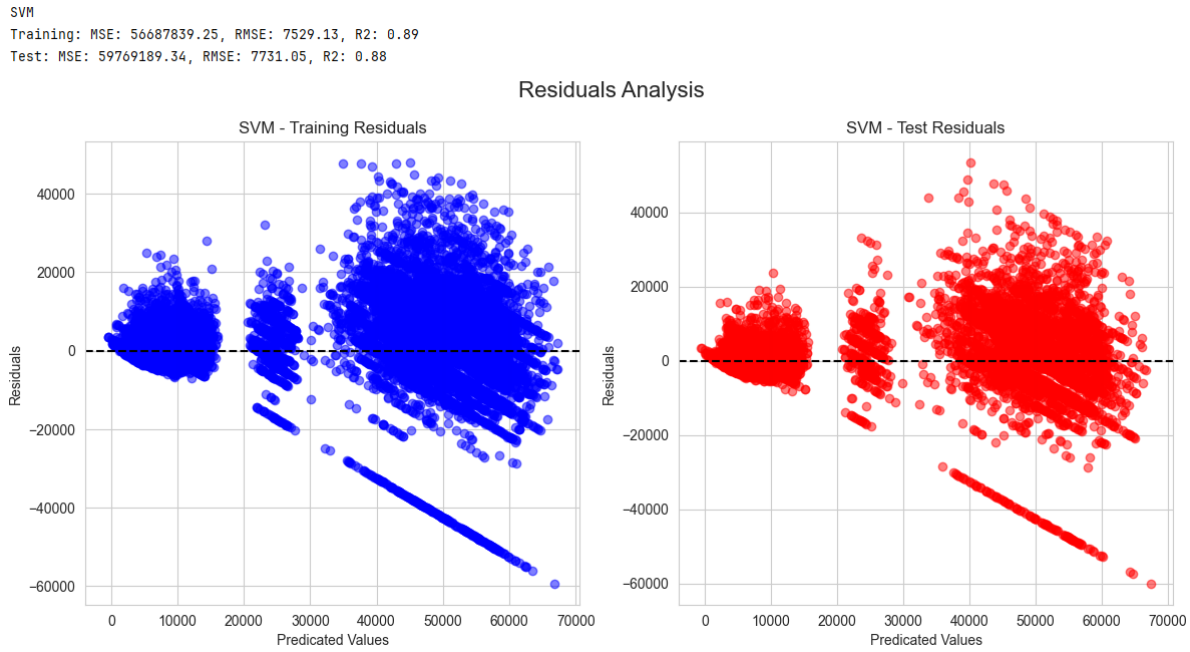
Obrázok 4: Rezíduá a Random Forest s príslušnými (R)MSE a R2 hodnotami

2.3 Support Vector Machine

Je to nelineárny model, ktorý sa pokúša nájsť optimálnu hranicu alebo "hyperrovinu"medzi triedami, aby maximalizoval margin (vzdialenosť) medzi rôznymi triedami v trénovacích dátach.

Ako jadro (typ funkcie, ktorá sa použije na transformáciu dát do vyššieho rozmeru, aby sa dala nájsť optimálna hyperrovina pre regresiu) bola použité **rbf** - veľmi silné pri modelovaní nelineárnych vzorcov v dátach.

Na obrázku nižšie môžeme vidieť vizualizáciu reziduí a taktiež vypísania (R)MSE a R2 skóre pre trénovaciu a testovaciu množinu. Ako základ pre porovnanie modelov budeme brať R(MSE) a R2 skóre pre testovaciu množinu.



Obrázok 5: Support Vector Machine s príslušnými (R)MSE a R2 hodnotami

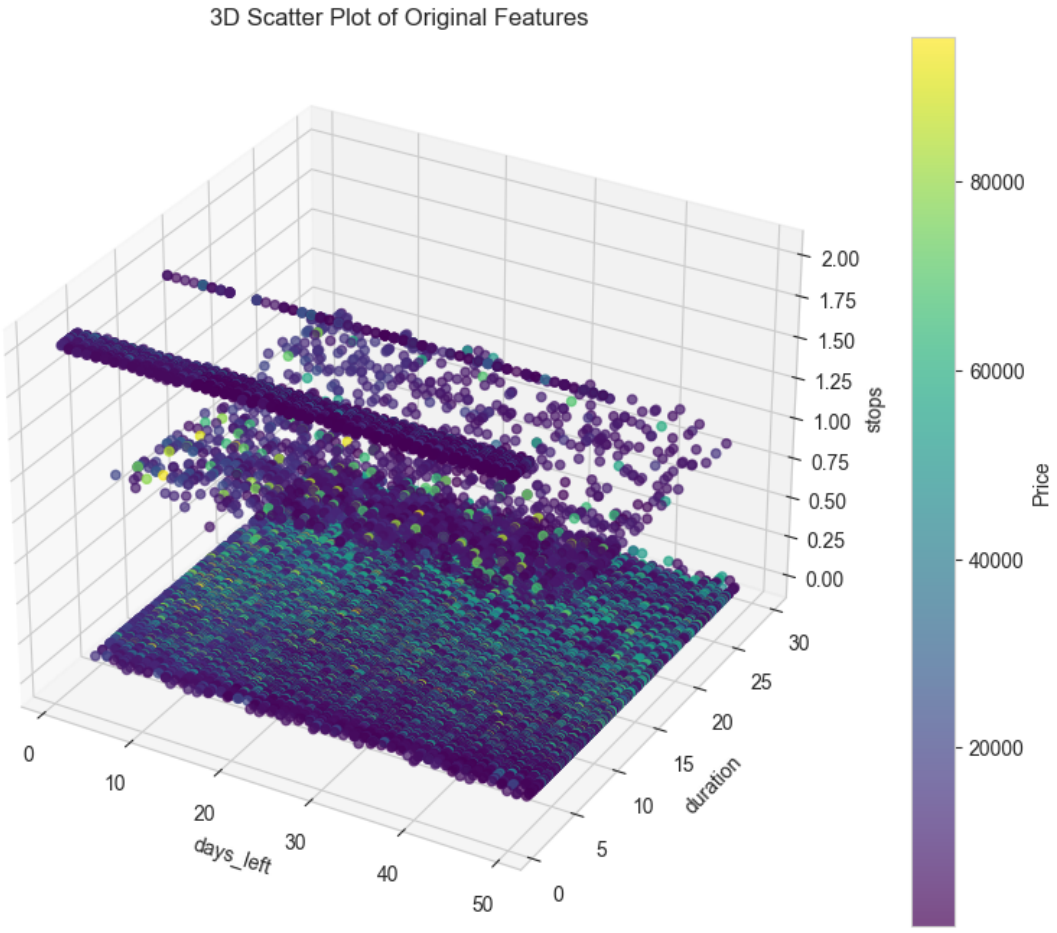
2.4 Porovnanie

Na záver tejto kapitoly budeme porovnávať tieto 3 modely. Ako je možné vidieť na základe (R)MSE a R2 skóre obstáli tieto modely nasledovne (od najlepšieho po najhorší), pričom hodnoty (R)MSE a R2 boli porovnávané vzhľadom na testovaciu množinu:

1. Stromový súborový model - Random Forest:
    - MSE: 38784499.18
    - RMSE: 6227.72
    - R2: 0.92
2. SVM
    - MSE: 59769189.34
    - RMSE: 7731.05
    - R2: 0.88
3. Rozhodovací strom
    - MSE: 67844716.99
    - RMSE: 8236.79
    - R2: 0.87

3 Redukcia dimenzie

Boli vybrané 3 príznaky - *days_left, duration* a *stops*, pričom tieto príznaky boli vybrané ešte pred normalizáciou. Pre tieto dáta bol vykreslený graf, na ktorom sú dáta vyfarbené podľa výstupného parametra - ceny (*price*). Tento graf umožňuje priamo vidieť vzťahy medzi vybranými premennými a cenou.

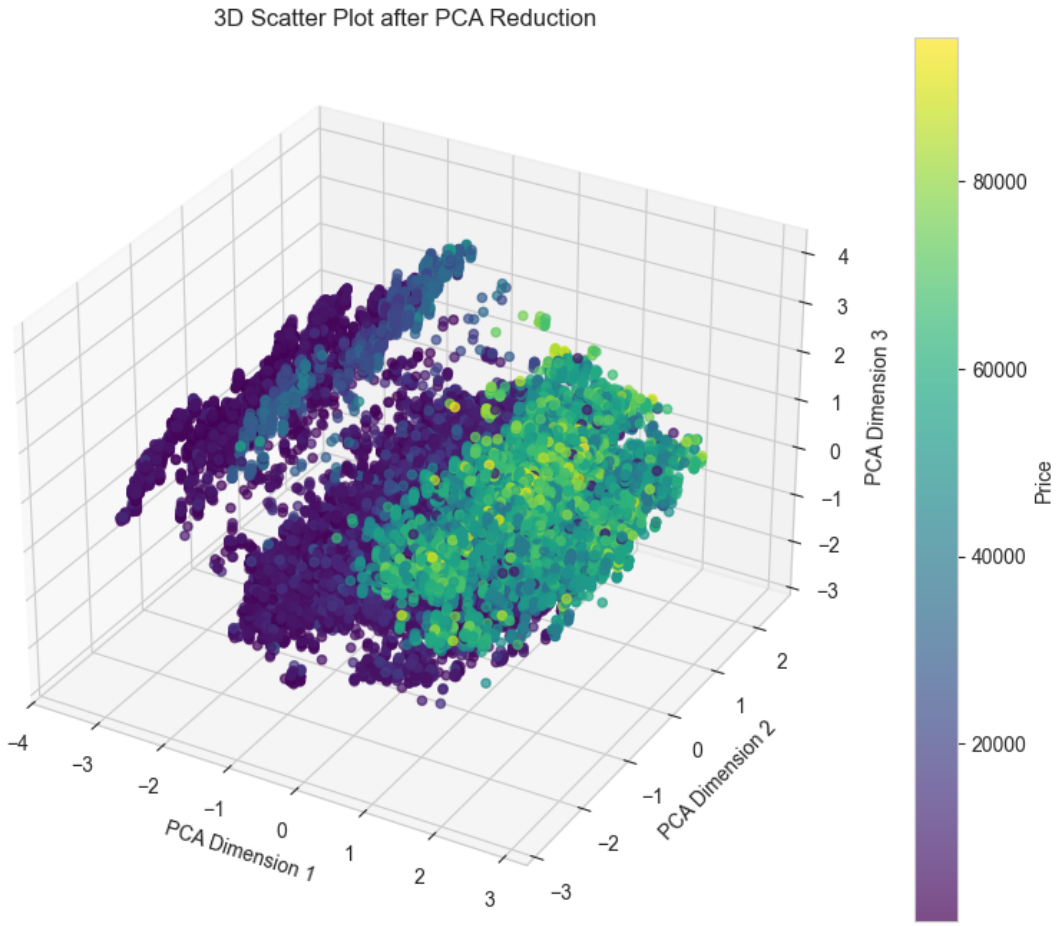


Obrázok 6: 3D Scatter Plot pre 3 pôvodné príznaky

Každý bod v grafe predstavuje jeden let s konkrétnymi hodnotami príznakov. Farebná škála napravo od grafu ukazuje, ako sú farebne kódované ceny. V tomto prípade prechádza od modrej (nižšia cena) po žltú (vyššia cena). Každý bod teda reprezentuje let, kde farba zobrazuje cenu. Let má konkrétne hodnoty príznakov (*days_left, duration, stops*).

Cieľom Principal Component Analysis (PCA) je znížiť počet atribútov (stĺpcov), pričom sa zachovávajú čo najviac informácií o variabilite dát, pričom sa zjednodušuje model a vizualizácia dát. PCA pracuje na princípe lineárnej transformácie pôvodných

množín do nových, ktoré sú korelované medzi sebou čo najmenej a zachovávajú čo najviac variability v dátach.



Obrázok 7: 3D Scatter Plot pre 3 príznaky po PCA redukcií a normalizácii

Pre tieto dáta bol opäť vykreslený graf, na ktorom sú dáta vyfarbené podľa výstupného parametra - ceny (*price*). V tomto grafe sa kždý z hlavných komponentov (PCA Dimension 1, PCA Dimension 2 a PCA Dimension 3) vypočíta ako lineárna kombinácia predošlých množín v dátovej množine *flight_data.csv*. Na tomto grafe môžeme vidieť plynulý prechod farieb, to znamená, že existuje vzťah medzi niektorými hlavnými komponentami a cieľovou premennou (price). To naznačuje, že PCA dokázalo zachytiť faktory ovplyvňujúce cenu. Môžeme taktiež vidieť. že sa bodky s podobnými farbami zoskupujú, môže to naznačovať, že v dátach sú skupiny (drahé a lacné letenky) s jasne odlíšiteľnými charakteristikami.

Tento graf ukazuje tie isté dáta redukované na hlavnú štruktúru, pričom zachováva kľúčové vzory a zoskupenia bez priamej interpretácie pôvodných premenných.

4 Trénovanie modelu pre zmenšenú množinu

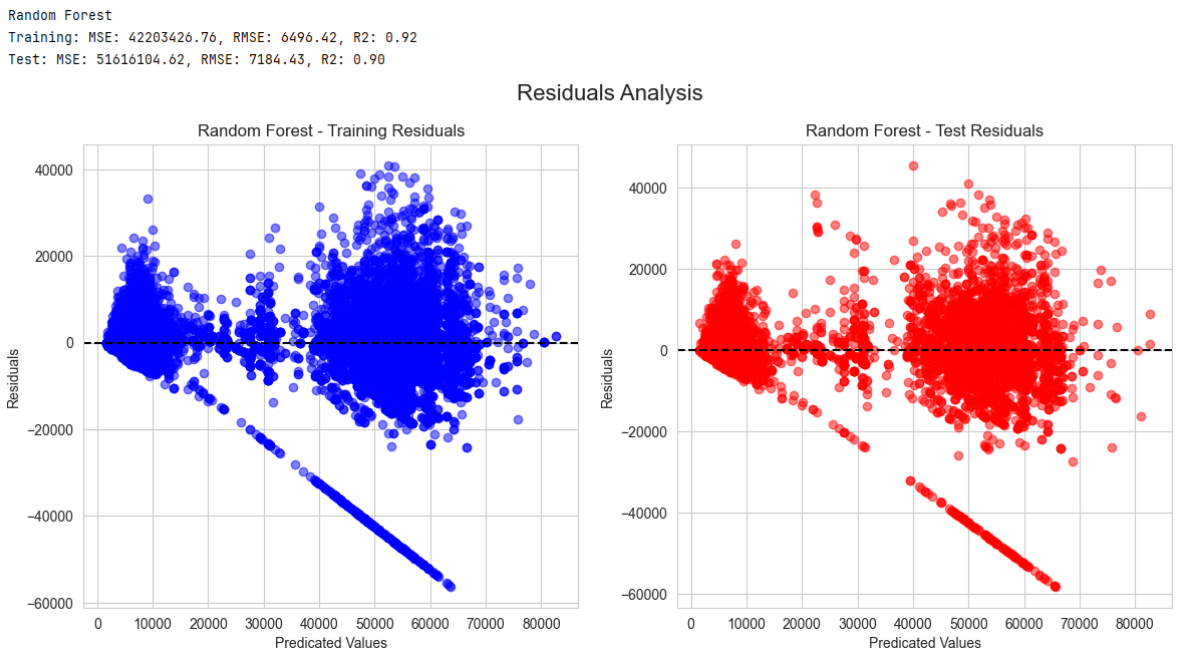
Pre toto trénovanie bol vybraný najúspešnejší model - stromový súborový model - Random Forest.

4.1 Korelačné príznaky

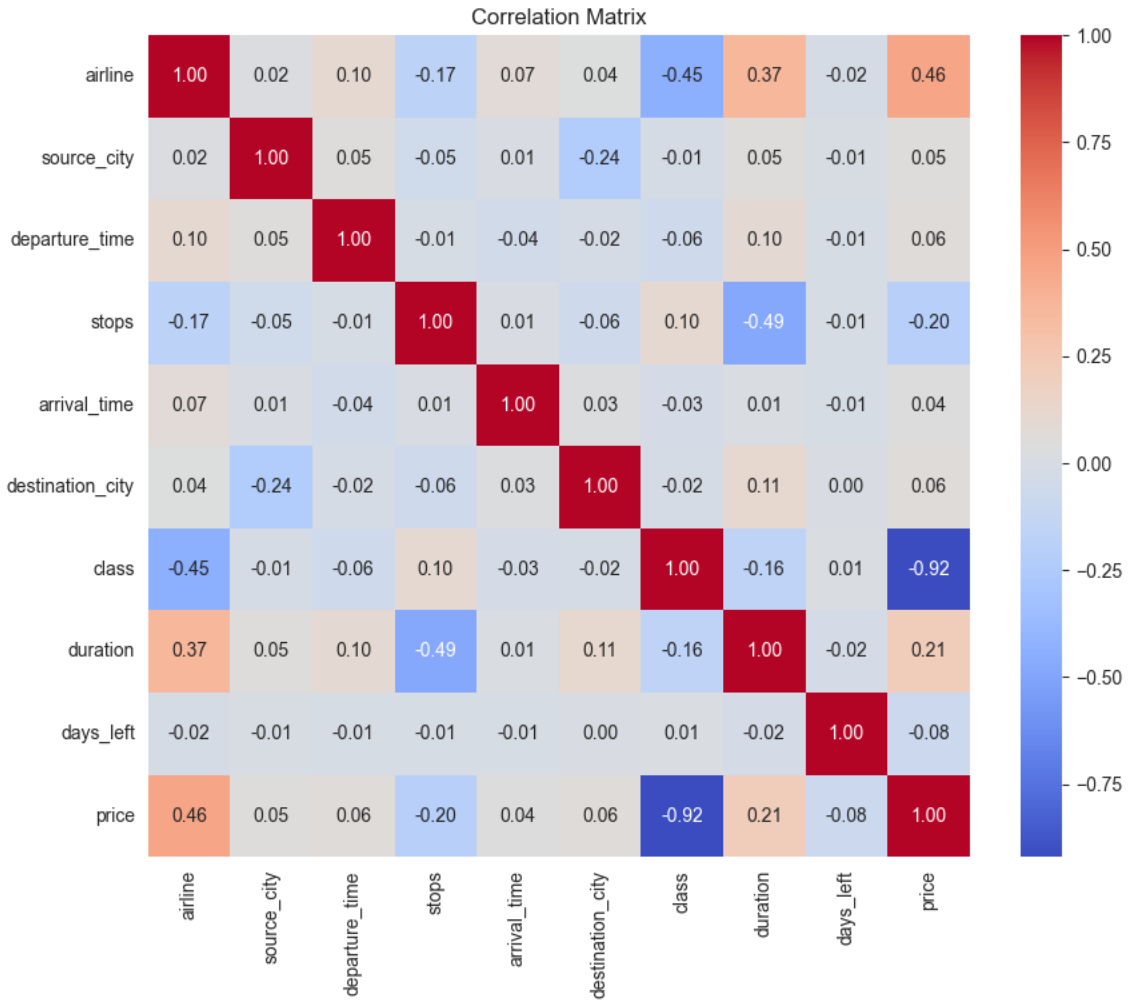
V tomto prípade bola vytvorená korelačná matica (bol použitý dataset so zakódovanými hodnotami), pričom boli vybrané vzťahy, ktoré korelujú s price, ak je ich (absolútna) hodnota väčšia ako 0.1. S týmito práznakmi bol následne vytorený zoznam (list), pričom bol odstránnený príznak price (keďže hodnota tejto korelácie v tomto prípade je vždy 1 - koreluje sám so sebou).

Vybrané príznaky sú:

- airline
- class
- duration
- stops



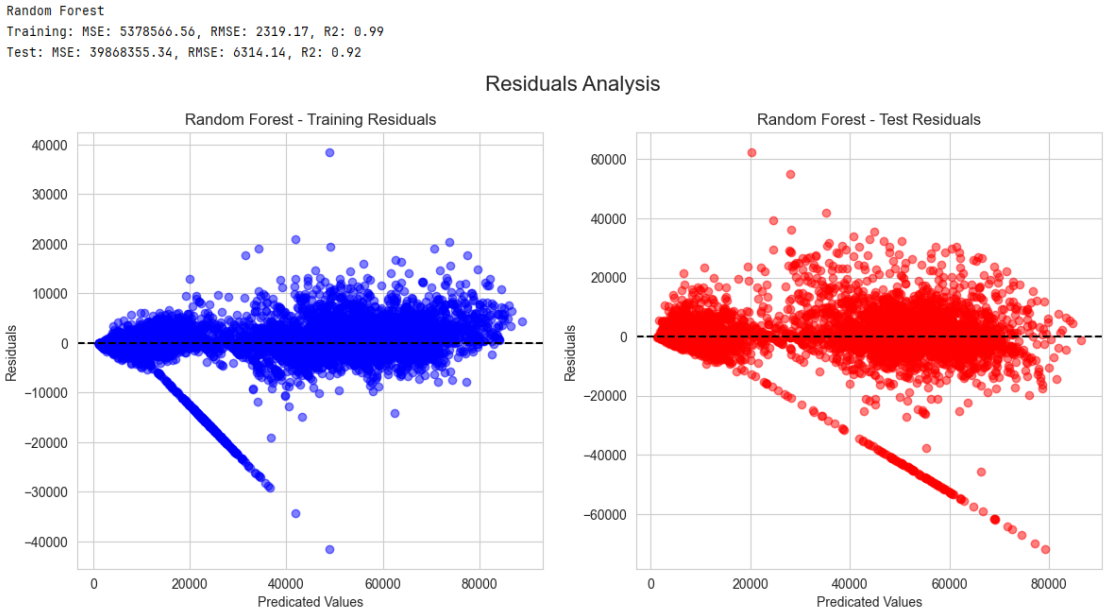
Obrázok 8: Rezídua na základe vybraných príznakov z korelačnej matice



Obrázok 9: Korelačná matica (zakódovaného) datasetu

4.2 Príznaky podľa dôležitosti

V tomto prípade sme pracovali s vybranými najdôležitejšími príznakmi, ktoré boli popísané v kapitole 2.2. Nižšie zobrazíme graf reziduí vrátane (R)MSE a R2 skóre.



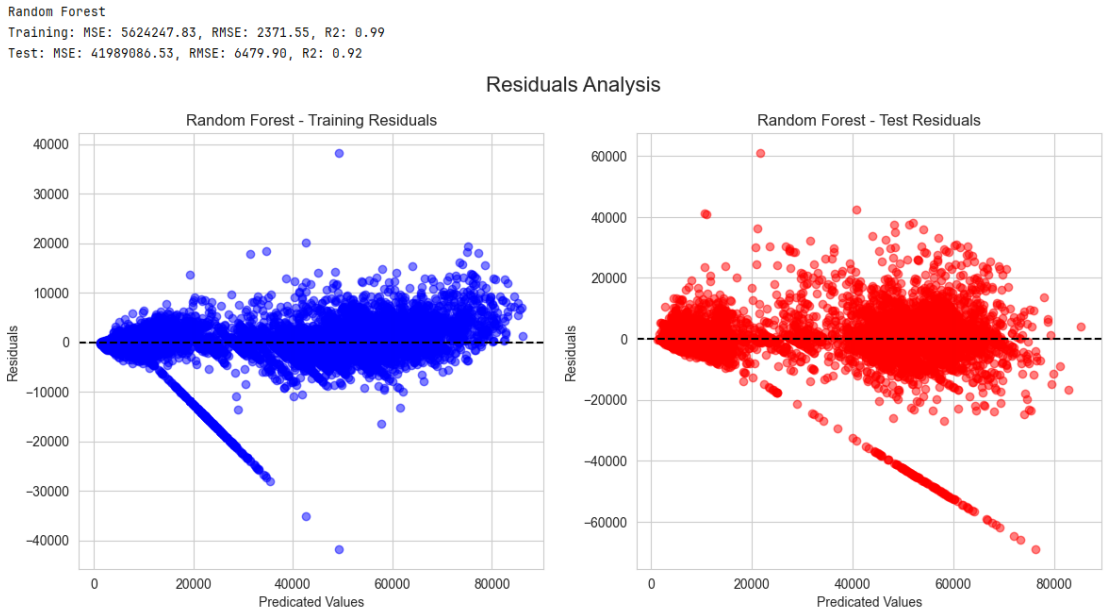
Obrázok 10: Reziduá na základe najdôležitejších príznakov v Random Forest modeli

4.3 Príznaky podľa PCA

Ako už bolo spomínane vyššie, PCA je technika na zníženie dimenzionality, ktorá transformuje dáta tak, že sa vyberajú najdôležitejšie komponenty (zachytávajúce najväčšiu variabilitu v dátach). Tento proces znižuje počet vstupných atribútov (príznakov), pričom sa snaží zachovať čo najviac informácie (variancie).

V tomto prípade sa zachová 95 % celkovej vysvetlenej variancie v dátach. Počet komponentov sa určí automaticky tak, aby sme zachovali túto percentuálnu hodnotu. PCA naučí najlepšie komponenty, ktoré zachytávajú najviac variancie v dátach. Transformuje pôvodné dáta (X_train) do nového priestoru definovaného týmito hlavnými komponentami. Tento krok zníži dimenzionalitu dát na menší počet komponentov, pričom sa zachová čo najviac informácie.

Na testovaciu množinu X_test aplikujeme rovnakú transformáciu ako na trénovaciu množinu, ale už bez fitovania (neprispôsobujeme PCA na testovacie dáta, iba ich transformujeme podľa PCA, ktoré sme naučili na trénovacích dátach).



Obrázok 11: Reziduá na základe PCA príznakov

4.4 Porovnanie

Na záver tejto kapitoly budeme porovnávať 3 prípady trénovania modelu Random Forest na základe rozličného výberu príznakov. Toto porovnanie je zobrazené nižšie, od najúspešnejšieho po namenej úspešný výber príznakov na základe (R)MSE a R2 skóre.

1. Výber príznakov podľa dôležitosti:
    - MSE: 39868355.34
    - RMSE: 6314.14
    - R2: 0.92
2. Výber príznakov podľa PCA
    - MSE: 41989086.53
    - RMSE: 6479.90
    - R2: 0.92
3. Výber príznakov na základe korelačnej matice
    - MSE: 51616104.62
    - RMSE: 7184.43
    - R2: 0.90

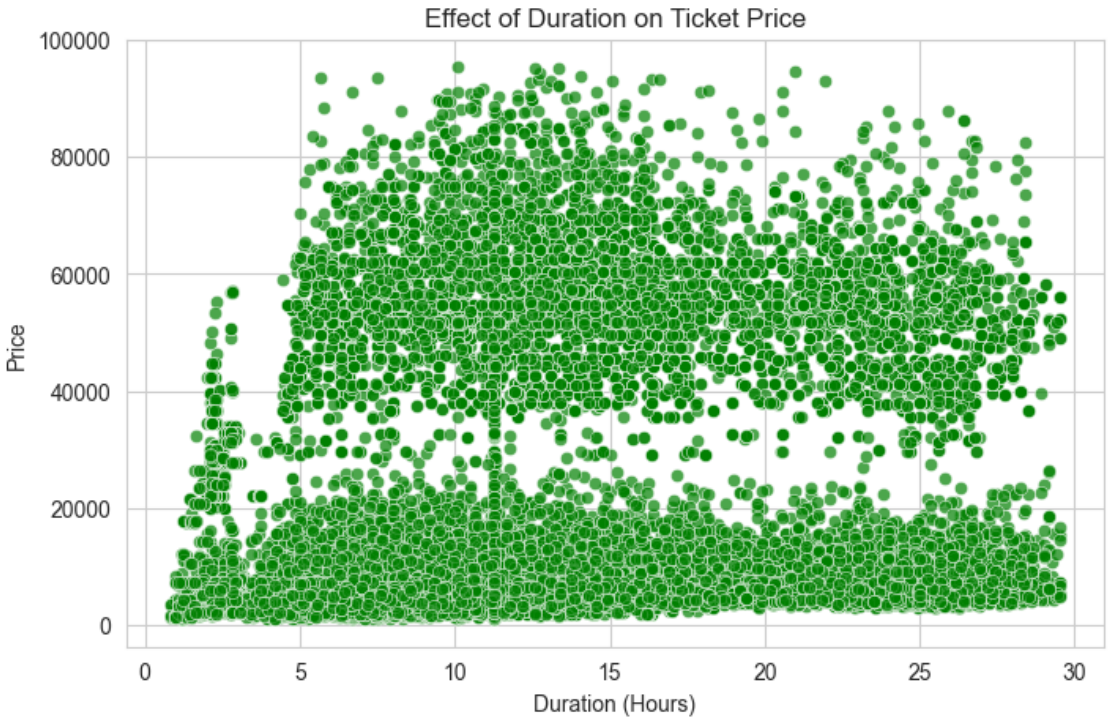
Na základe porovnania s modelmi v sekcii 2.4 môžeme konštatovať, že z celkového hľadiska nastalo zlepšenie pri (R)MSE a R2 skóre. Pri najúspešnejšom výbere (na základe príznakov) nastalo miernejšie 'zhoršenie' v (R)MSE skóre - to je však zanedbateľné.

5 Analýza dát

Výstupné dáta boli analyzované pomocou Exploratory Data Analysis (EDA). V tomto prípade pracujeme s dátami (po zakódovaní - aby bolo možné zobraziť viaceré príznaky, keďže korelačná matica učruje vzťah medzi číselnými hodnotmai), ktoré neobsahujú outliery, chýbajúce hodnoty a identifikátory. Tieto hodnoty boli vybrané na základe korelačnej matice, ktorá je znázornená na obrázku 9.

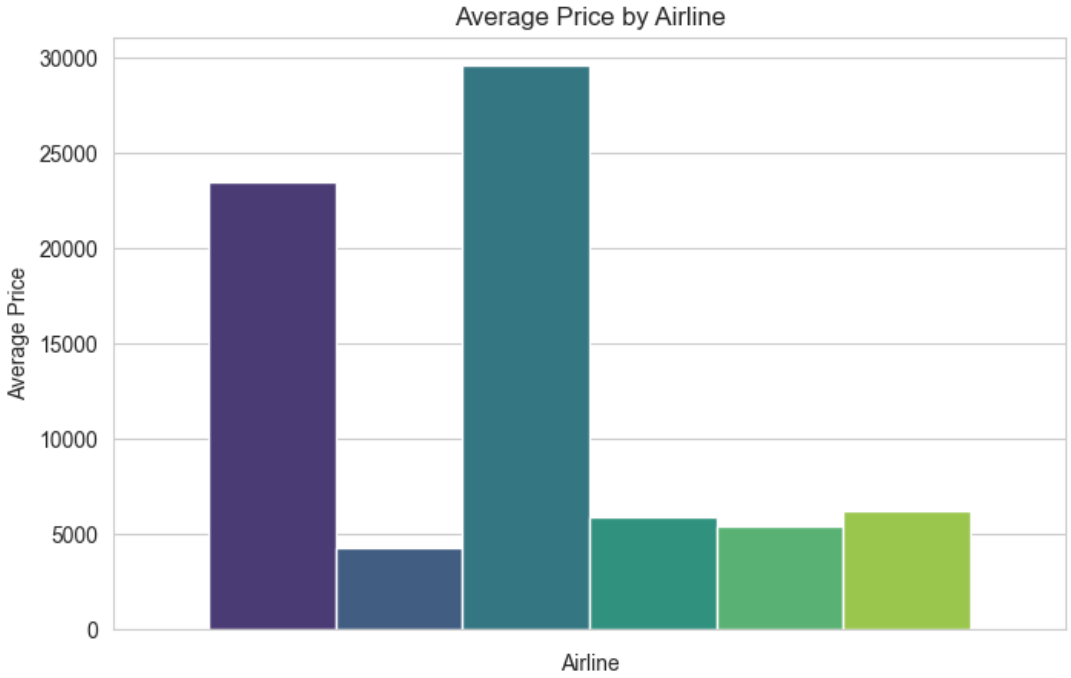
Obrázok 12: Cena vs. Trieda

Tento graf (violinplot - husľový graf) zobrazuje distribúciu cien pre rôzne triedy (napr. economy, business). Tvar a šírka grafu ukazujú hustotu cien, pričom rozdiely medzi triedami indikujú, ako sa cena mení v závislosti od triedy letu.



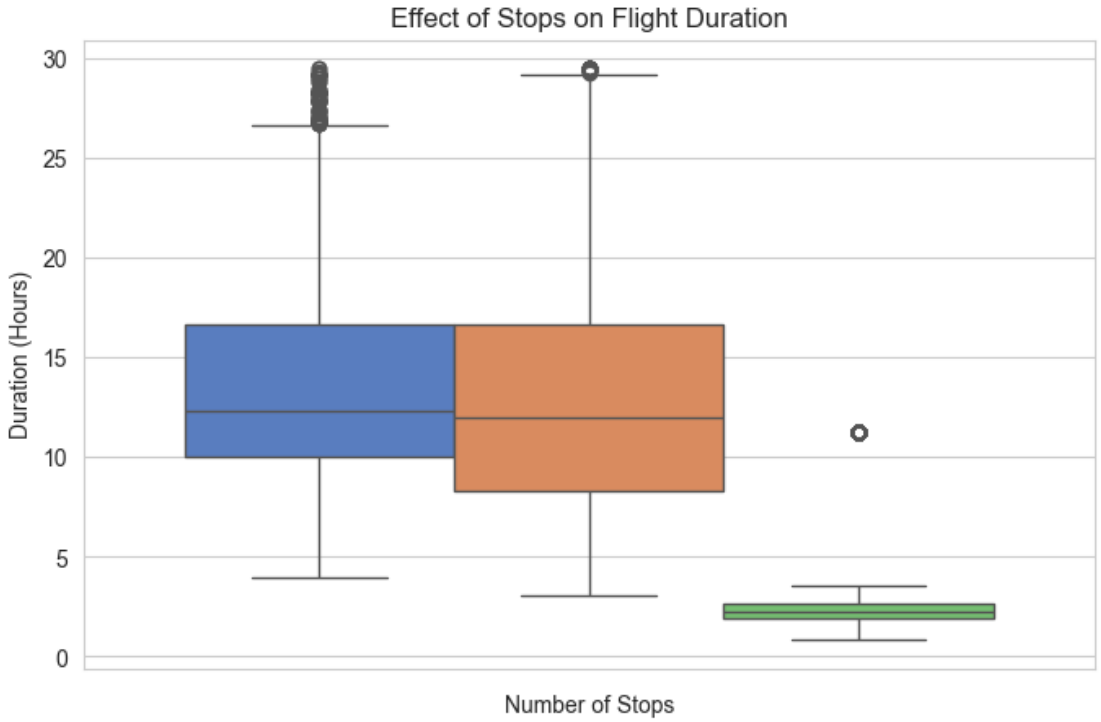
Obrázok 13: Trvanie letu vs. Cena

Tento graf (scatter plot) ukazuje vzťah medzi trvaním letu (v hodinách) a cenou. Bodky predstavujú jednotlivé lety, ich pozícia ukazuje trvanie a zodpovedajúcu cenu. Trendy alebo vzory môžu indikovať, že dlhšie lety sú často drahšie.



Obrázok 14: Cena vzhľadom na leteckú spoločnosť

Tento graf (barplot) porovnáva priemerné ceny jednotlivých leteckých spoločností. Výška stĺpcov zobrazuje priemernú cenu pre každú spoločnosť, čím umožňuje rýchlo identifikovať, ktorá spoločnosť má vyššie alebo nižšie priemerné ceny.



Obrázok 15: Počet zastávok vs. Trvanie letu

Tento graf (boxplot) vizualizuje, ako počet zastávok ovplyvňuje trvanie letu. Boxploty zobrazujú rozsah, medián a extrémy trvania pre rôzne počty zastávok (napr. bez zastávok, jedna zastávka, dve a viac zastávok).

Obrázok 16: Počet dní do odletu vs. Cena

Tento graf (regplot - scatter plot s regresiou) ukazuje vzťah medzi počtom dní zostávajúcich do odletu a cenou letu, s regresnou čiarou na identifikáciu trendu. Bodky predstavujú lety a ich ceny, zatiaľ čo červená čiara ukazuje, ako cena typicky klesá/stúpa so zmenšujúcim/zvyšujúcim sa počtom dní do odletu.

6 Neurónová sieť

V tejto kapitole bude popísaná neurónová sieť pre úlohu regresie (cieľový výstup je spojitá hodnota[8] - price). Vytvorenie neurónovej siete so zvolenými vstupnými parametrami (vstupná trénovacia množina, 1. vrstva neurónov, 2. vrstva neurónov a k ním prislúchajúce pravdepodobnosti vypnutia určitých neurónov (30 % - aby bola sieť robustnejšia a nepretrénovala sa) a rýchlosť učenia) zabezpečuje funkcia *create_network()*.

Architektúra siete:

1. *Vstupná vrstva:* Počet vstupných neurónov sa rovná počtu príznakov vo vstupných trénovacích dátach. Táto vrstva očakáva, že každý vzor bude reprezentovaný ako vektor dĺžky rovnej počtu príznakov.

2. *1. Skrytá vrstva:* Náhodne vypína 40% neurónov počas trénovania, aby sa znížila možnosť pretrénovania modelu. Obsahuje 64 neurónov a používa ReLU ako aktivačnú funkciu.

3. *2. Skrytá vrstva:* Náhodne vypína 40% neurónov počas trénovania, aby sa znížila možnosť pretrénovania modelu. Obsahuje 32 neurónov a používa ReLU ako aktivačnú funkciu.

4. *3. Skrytá vrstva:* 30% neurónov počas trénovania, aby sa znížila možnosť pretrénovania modelu. Obsahuje 32 neurónov a používa ReLU ako aktivačnú funkciu.

5. *Výstupná vrstva:* Obsahuje 1 (výstupný) neurón, ktorý predpovedá spojitú hodnotu. Použitie lineárnej aktivačnej funkcie je vhodné pre regresné úlohy.

Optimalizácia:

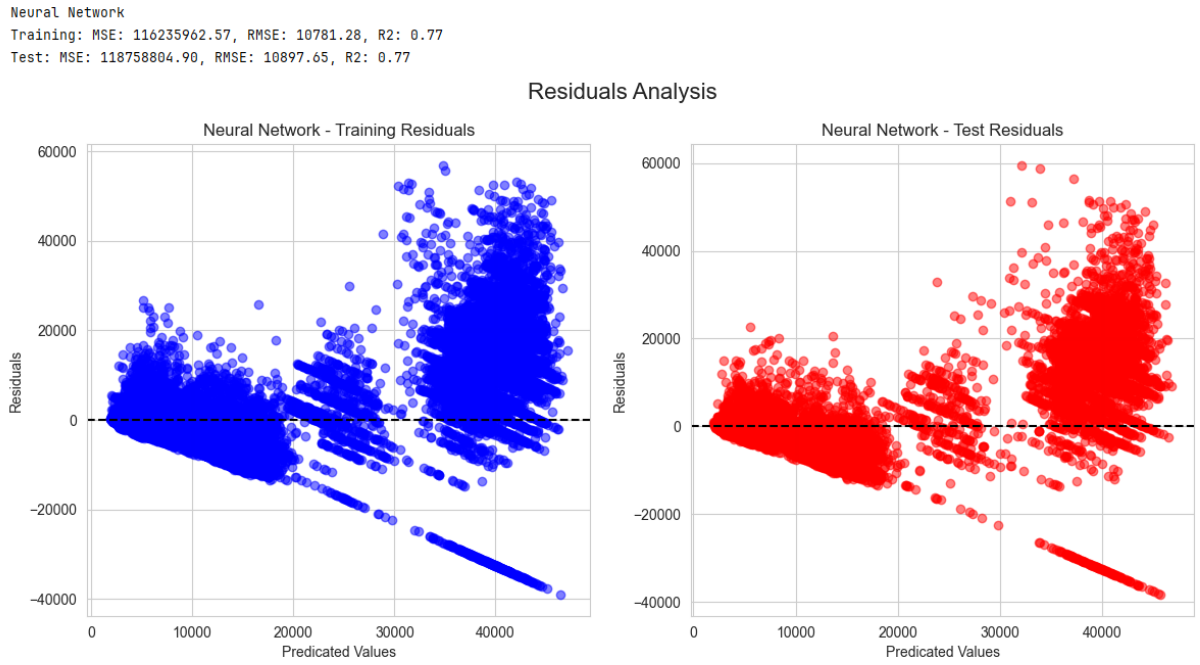
- *Optimalizátor:* Adam - Adaptívny algoritmus optimalizácie s počiatočnou rýchlosťou učenia nastavenou na 0.0001.

- *Stratová funkcia:* Používa sa na minimalizáciu priemernej kvadratickej chyby (MSE), čo je štandardná metrika pre regresiu.

- *Mean Squared Errror:* Sledovanie priemernej kvadratickej chyby počas trénovania a validácie.

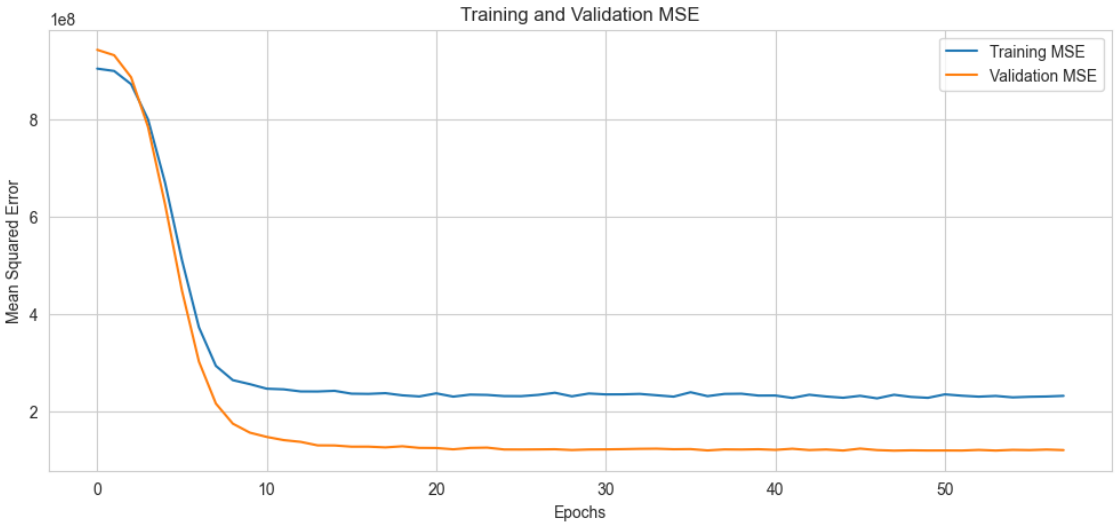
[8]Typ výstupu, ktorý môže byť akékoľvek reálne číslo v danom intervale

V tomto modeli je použitých 30% dát na valudáciu (rozdelenie časti tréningovej množiny na validačnú množinu počas trénovania modelu, aj keď nie je výslovne poskytnutá validačná množina.), pričom sieť sa učí počas 100 epôch a vzory sa spracovávajú po 32 dávkach. Taktiež bol použitý EarlyStopping aby sa vyplo trénovanie siete v prípade, že sa sieť pretrénovava (zastaví trénovanie, ak sa nezlepčí po 10-tich epochách).



Obrázok 17: Rezídua a neurónová sieť

Tieto grafy zobrazujú (R)MSE a R2 skóre pre neurónovú sieť s vyššie uvedenou architektúrou.



Obrázok 18: MSE trénovania a validácie neurónovej siete

Záver

Cieľom zadania bolo implementovať program, ktorý predpovedá cenu letenky, pričom vstupné dáta boli poskytnuté z AIS.

Dáta boli očistené od identifikátorov, nulových hodnôt a duplicít, pričom v prípade numerických množín bola vykonaná taktiež aj kontrola, či sa v týchto množinách nenachádza aj textový reťazec a ak áno, nech sa nahradí mediánom zvyšných hodnôt. Tieto dáta boli následne zakódované LabelEncoding-om a TargetEncoding-om.

Následne boli dáta rozdelené na vstupnú a výstupnú trénovaciu a testovaciu množinu, pričom vstupná množina bola normalizovaná StandardScaler-om. Ďalej boli natrénované 3 modely - rozhodovací strom, Random Forest a SVM, pričom tieto modely boli vyhodnotené na trénovacej a testovacej množine pomocou (R)MSE a R2 skóre.

Ďakej boli vybrané 3 príznaky pred normalizáciou, ktoré boli na grafe zafarbené v závislosti od výstupného parametra (ceny). Potom bola minimalizovaná množina pomocou PCA na 3 dimenzie a dáta boli na grafe opäť zafarbené podľa výstupného parametra (ceny).

Následne boli vybrané 3 podmnožiny príznakov (podľa korelačnej matice, podľa dôležitosti príznakov z Random Forest modelu a podľa variance pomocou PCA).

Záver tejto práce sa týkal doplňujúcich úloh pre analýzu dát pomocou EDA(vykreslenie piatich rôznych grafov pre päť rôznych závislostí) a následného trénovania neurónovej siete per regresné úlohy.