SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

AIS ID: 92320

NEURÓNOVÉ SIETE - KATEGORIZÁCIA POČASIA

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
UV	Ultrafialové žiarenie
X	Vstupná množina
Y	Výstupná množina

Úvod

Cieľom tohto zadania je natrénovať neurónovú sieť pre kategorizáciu počasia, pričom pracujeme s dátami z AIS. Najprv bolo tieto dáta potrebné načítať z príslušného súboru a následne boli upravené tak, aby neobsahovali žiadne chýbajúce alebo neobvyklé hodnoty (outliery), Ak daná množina (resp. stĺpec v tabuľke dát) obsahoval príliš veľa chýbajúcich hodnôt (viac, ako 25 % dát chýbalo), bol tento stĺpec odstránený. Dáta boli následne vhodne zakódované, škálované a rozdelené do množín, aby bolo možné natrénovať jednoduchý klasifikačný model. Boli vykonané 4 experimtny, pričom pre najlepší z nich bola vytvorená tabuľka a konfúzna matica.

Následne boli dáta analyzované pomocou Exploratory Data Analysis (EDA), pričom boli použité dáta už upravené dáta, očistené od outlierov, duplicít a chýbajúcich hodnôt. Následne bolo vykreslených a popísaných 5 rôznych grafov, na základe korelačnej matice, ktorá ukázala, ako jednotlivé hodnoty medzi sebou vzájomne korelujú (zisťoval sa ich vzájomný vzťah, ako spolu dané hodnoty súvisia).

Následne bola natrénovaná neurónová sieť, pričom bolo vykonaných 5 pokusov, kde boli zmenené rôzne parametre, aby sme zistili rozdiely v jednotlivých experimentoch pri rôznych hodnotách. Konfigurácie týchto experimentov boli zapísané v jednej tabuľke, pričom pre najlepšie a najhoršie trénovanie bola vykreslená konfúzna matica.

1 Vstupné dáta

V tejto kapitole bude popísaný spôsob predspracovania a následného vyhodnotenia vstupných dát zo súboru *weather_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.
- Seabornv[3]: Vysokoúrovňová vizualizácia dát so zameraním na štatistiku.
- Matplotlib.pyplot[4]: Základná knižnica na tvorbu grafov s nízkoúrovňovou kontrolou nad vizualizáciami.
- Tensorflow (Keras)[5]: Tvorba a trénovanie modelov strojového učenia a neurónových sietí. API Keras zjednodušuje tvorbu a trénovanie neurónových sietí.
- skLearn[6]: Knižnica určená pre klasické metódy strojového učenia. Obsahuje množstvo algoritmov pre klasifikáciu, regresiu,...

1.1 Predspracovanie dát

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 odstránené množiny[7], v ktorých chýbalo viac ako 25% dát (v tomto prípade to bola množina Ïrradiance"). Potom sme rozdelili dáta do dvoch skupín - tie, ktorý by mali obsahovať len číselné hodnoty a tie, ktoré by mali obsahovať len textové hodnoty.

Množiny dát, ktoré mali obsahovať len číselné hodnoty, boli prekontrolované funkciou *median_replace()*, ktorá v prípade výskytu napr. textových hodnôt túto hodnotu zmazala a nahradila mediánom všetkých hodnôt v danej množine. Pre odstránenie outlierov (nezvyčajných hodnôt) bola navrhnutá funkcia *outliers()*, ktorá vrátila novú dátovú

[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://www.tensorflow.org/guide/keras
[6] https://scikit-learn.org/stable/
[7] V tabuľke dát sú to stĺpce

množinu bez outlierov[8] a následne boli odstránené duplicitné hodnoty. Proces očistenia dát bol zakončený kontrolou, ktorá vrátila stĺpce (množiny) dát podľa nasledujúcich intervalov:

- Temperature: $-25°C$ do $109°C$
- Humidity: od 20% do 109%
- Wind Speed: 0 km/h do 48,5 km/h
- Precipitation (%): od 0% do 109%
- Cloud Cover: partly cloudy, clear, overcast, cloudy
- Atmospheric Perssure: od 984 hPa do 1067 hPa
- UV Index: od 0 do 14
- Season: Winter, Spring, Summer, Autumn
- Visibility (km): od 0 km do 20 km
- Location: inland, mountain, coastal
- Weather Type: Rainy, Cloudy, Sunny, Snowy
- Irradiance: od 200 do 800 W/m (**stĺpec(množina) odstránená z dôvodu veľkého počtu výskytov prázdnych hodnôt**)

Kódovanie nečíselných hodnôt prebehlo pomocou *Label Encoding*, ktorý priraďuje textovým hodnotám práve jedno číslo (napríklad množine dát "Weather Type", ktorá obsahuje hodnoty "Sunny, Cloudy, Rainy, Snowy") boli pridelené hodnoty ("0, 1, 2, 3"). Keďže účelom implementácie bola kategorizácia počasia, výstupná množina Y obsahovala práve jednu množinu dát - Weather Type. Vstupná množina X obsahuje zvyšné množiny dát.

Následne bola navrhnutá funkcia *train_model()*, ktorá vykonáva rozdelenie vstupných a výstupných dát (trénovacia[9], validačná[10] a testovacia[11] množina podľa zvoleného deliaceho pomeru). Následne je vykonaná normalizácia dát pomocou **MinMaxScaler**, ktorý škáluje, resp. normalizuje hodnoty vstupných dát (X) do intervalu (napríklad $< 0, 1 >$). Trénovanie modelu prebieha logistickou regresiou so zadaným počtom iterácii (v tomto prípade 200). Následne prebehne predikcia a vyhodnotenia dát, pričom výstupom tejto funkcie sú nasledovné veličiny:

- model logistickej regresie,
- presnosť modelu vzhľadom na testovacie dáta,

[8] Hodnoty, ktoré sa výrazné líšia od ostatných hodnôt v danej množine

[9] Model sa z týchto dát učí, ako predikovať cieľové hodnoty

[10] Slúži na optimalizáciu modelu počas tréningu (ladenie hyperparametrov - napr. rýchlosť učenia, počet vrstiev v neurónovej sieti, ...)

[11] Slúži na finálne vyhodnotenie modelu po tréningu a ladení hyperparametrov

- konfúzna matica.

Výstup tvoria taktiež aj normalizované dáta (tréningové, testovacie a validačné) a cieľové (výstupné) dáta.

Číslo experimentu	Deliaci pomer 1	Deliaci pomer 2	Úspešnosť
1	0.3	0.4	0.933941
2	0.3	0.1	0.936364
3	0.2	0.4	0.943052
4	0.1	0.2	0.963636

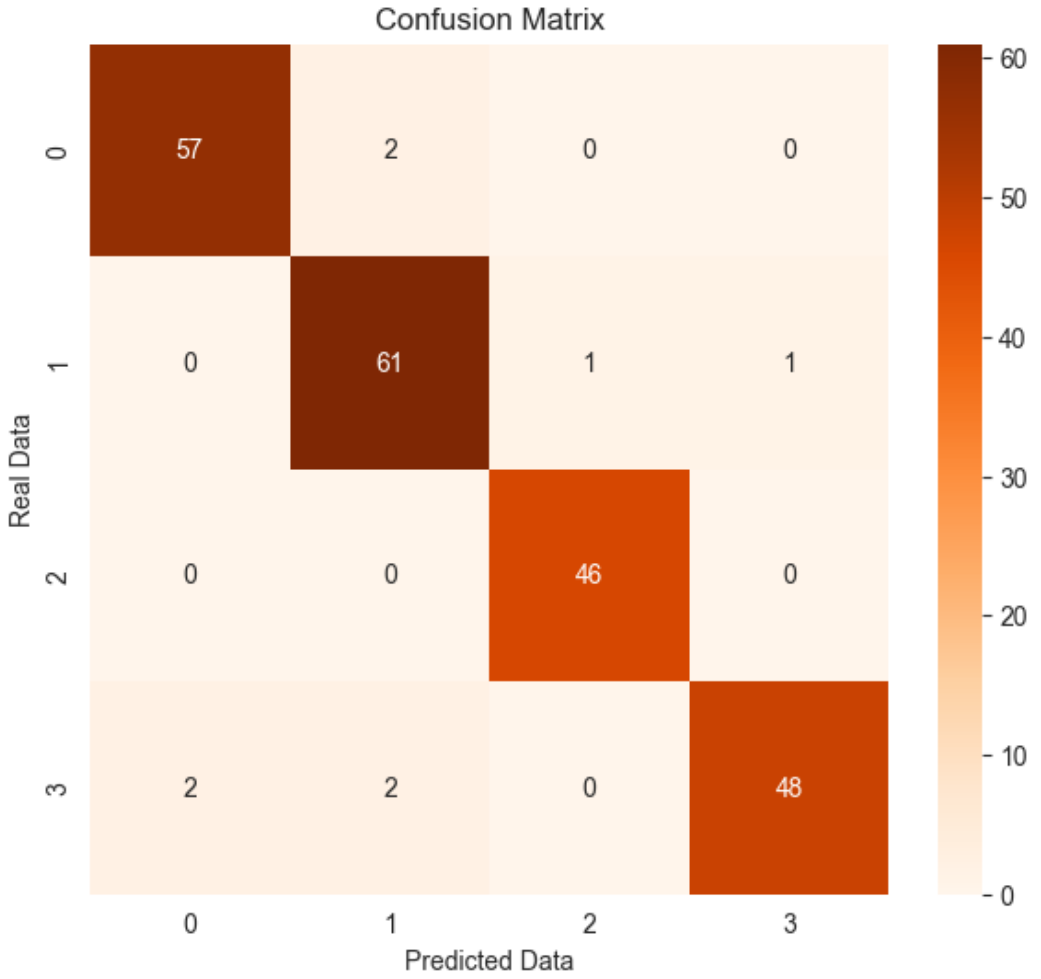
Tabuľka 2: Tabuľka vykonaných experimentov na základe pomeru trénovacích a testovacích dát s výslednou úspešnosťou

Ako je možné z tabuľky vidieť, najlepší (najúspešnejší) experiment je ten, ktorého deliaci pomer[12] je experiment číslo 3. Táto presnosť bola vygenerované funkciou *accuracy_score()* na základe skutočných hodnôt (ktoré obsahuje náš dataset) a predikovaných hodnôt (výstupné hodnoty - predpovede, vytvorené na základe vstupných dát).

Na obrázku nižšie je možné vidieť vygenerovanú konfúznu maticu[13]. Matica bola graficky vykreslená funkciou *generate_confusion_matrix()* na základe vygenerovanej matice *confusion_matrix* vo funkcii *train_model()*.

[12]Pomer trénovacích a testovacích dát, zvolená hodnota uvádza, aká časť dát bude použitá ako testovacie dáta (napr. $0.1 - 10\%$ pre testovacie dáta)

[13]Matica, ktorá znázorňuje výkonnosť klasifikačného modelu tým, že porovnáva predikované výsledky modelu s reálnymi (skutočnými) hodnotami. Hlavná diagonála obsahuje počet správne klasifikovaných predikcií (model správne identifikoval triedy

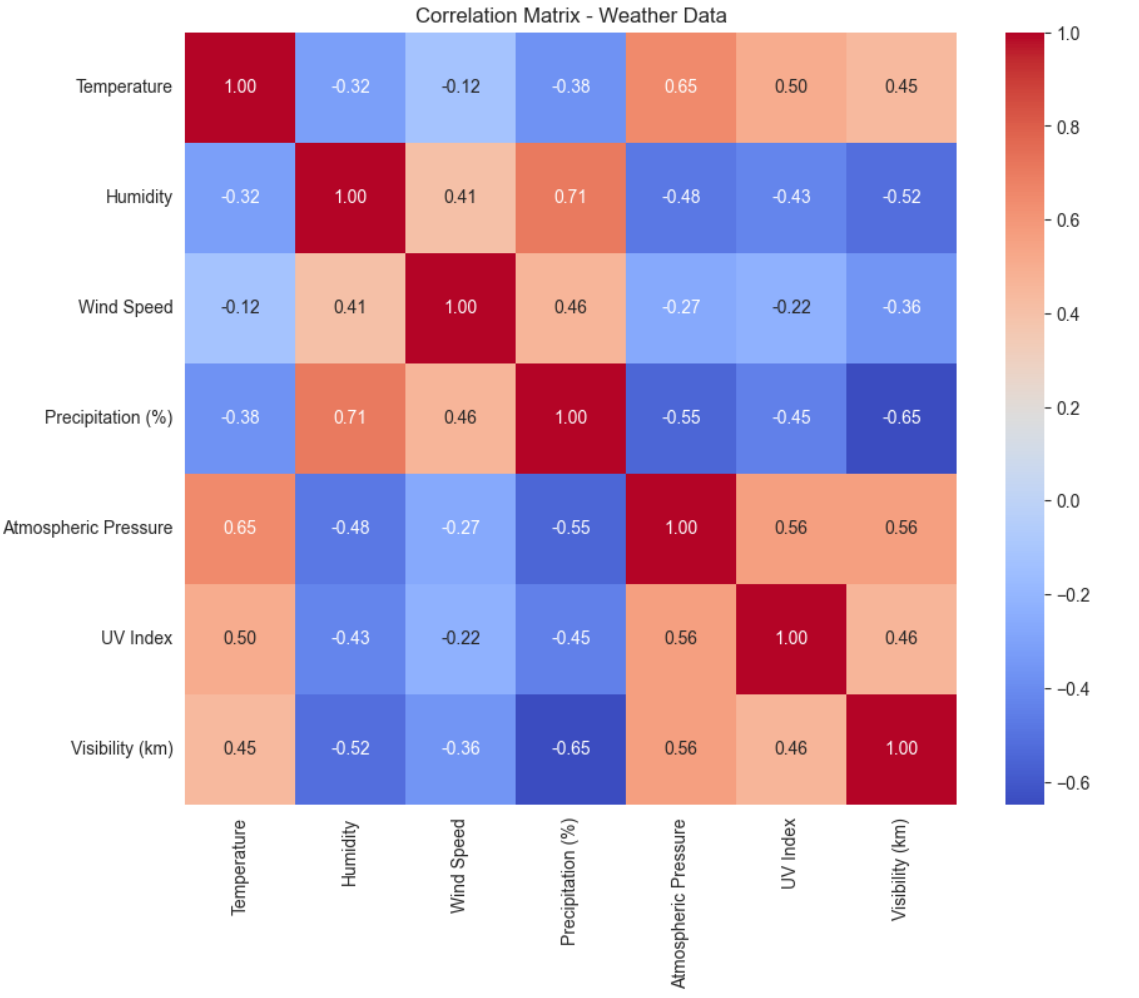


Obrázok 1: Zobrazenie konfúznej matice pre číselné hodnoty v dátovej množine, po odstránení nulových hodnôt a outlierov

2 Analýza dát

Výstupné dáta boli analyzované pomocou **Exploratory Data Analysis (EDA)**. V tomto prípade pracujeme s dátami (pred zakódovaním - aby bolo možné použiť slovné hodnoty), ktoré neobsahujú outliery alebo chýbajúce hodnoty. Zaujímavé hodnoty môžeme nájsť aj pomocou **korelačnej matice**[14]. Táto matica obsahuje zvyčajne hodnoty v intervale $[-1, 1]$:

- 1: Perfektná pozitívna korelácia (keď hodnota jednej premennej stúpa, druhá stúpa rovnako).
- -1: Perfektná negatívna korelácia (keď hodnota jednej premennej stúpa, druhá klesá).
- 0: Žiadna lineárna korelácia (premenné nie sú lineárne závislé).

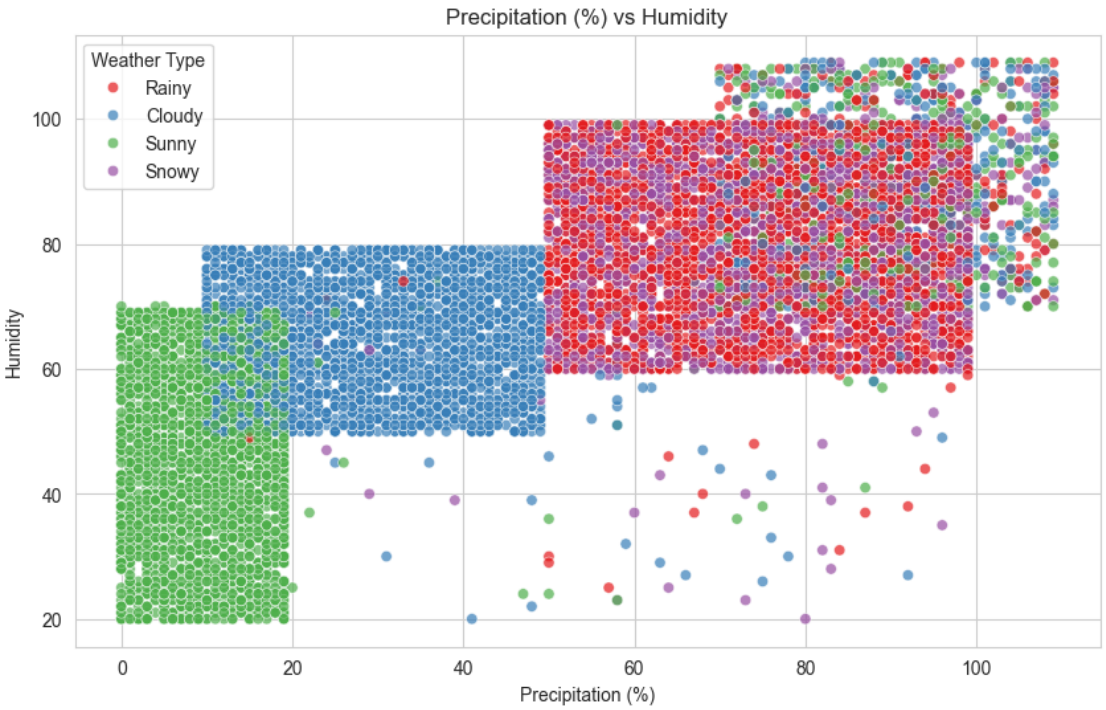


Obrázok 2: Zobrazenie korelačnej matice pre EDA

[14]Matica, ktorá ukazuje vzťahy medzi rôznymi premennými v dátových množinách

Táto matica bola vykreslená funkciou *generate_correlation_matrix()* pre množiny dát, v ktorých sa nachádzajú číselné hodnoty (keďže korelačná matica dokáže pracovať len s takými). Čím sa hodnota jednotlivých vzťahov približuje čo najbližšie ku 1 alebo -1, tým je korelácia lepšia.

1. **Pravdepodobnosť zrážok a vlhkosť:** Na tomto grafe znázorňuje os X pravdepodobnosť zrážok a os y percentuálny podiel vlkosti v ovzduší. Každý bod v grafe predstavuje jedinečný záznam (riadok) v súbore údajov, kde je záznamom kombinácia percenta zrážok a vlhkosti pre daný typ počasia.

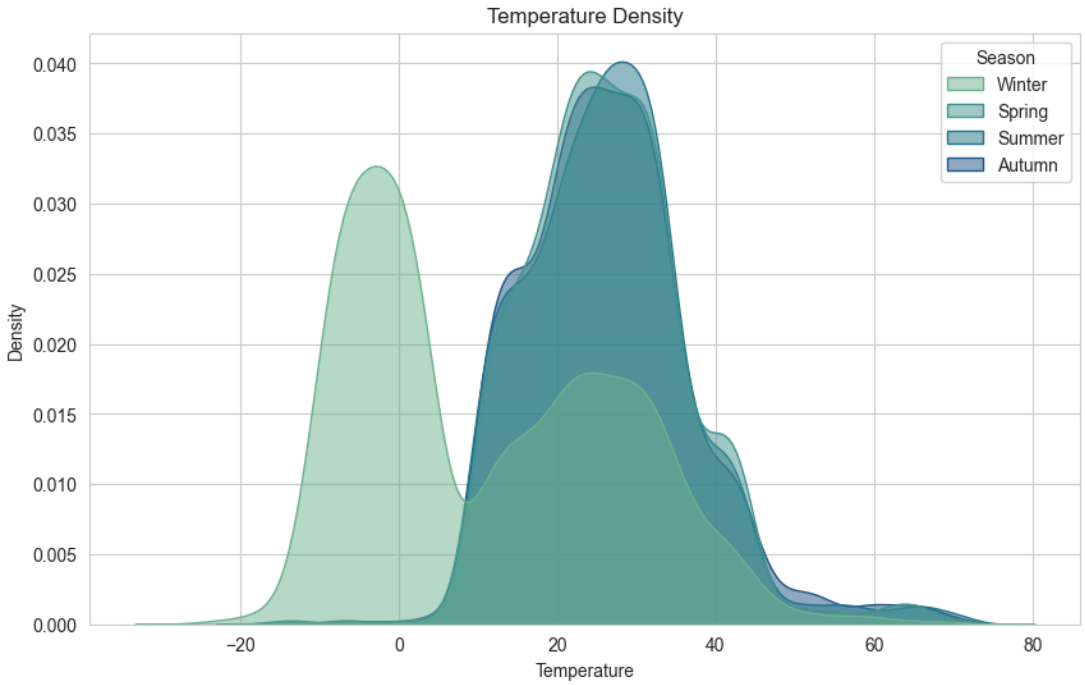


Obrázok 3: Zobrazenie grafu (scatterplot) pre pravdepodobnosť zrážok a vlhkosť

- Sunny: Tieto body môžu byť zvyčajne umiestnené v dolnej časti grafu, kde sú zrážky nízke (0-20 %) a vlhkosť tiež môže byť relatívne nízka.
- Rainy: Tieto body sa pravdepodobne nachádzajú v oblasti vyšších percentuálnych zrážok (prevažne 50-100 %) a môžu mať variabilnú vlhkosť, s niektorými bodmi na vyššej vlhkosti.
- Snowy: Podobne ako rainy, ale môžu mať nižšie teploty, a teda aj variabilitu vo vlhkosti.
- Cloudy: Tieto body môžu ukazovať stredné úrovne zrážok a vlhkosti, s možným výskytom nízkych až stredných hodnôt.

2. **Teplotná hustota vzhľadom na ročné obdobie:** Tento graf zobrazuje hustotu pravdepodobnosti teploty v závislosti od ročného obdobia. KDE plot ukazuje, ako je rozloženie teplôt rozšírené v rôznych obdobiach roka.

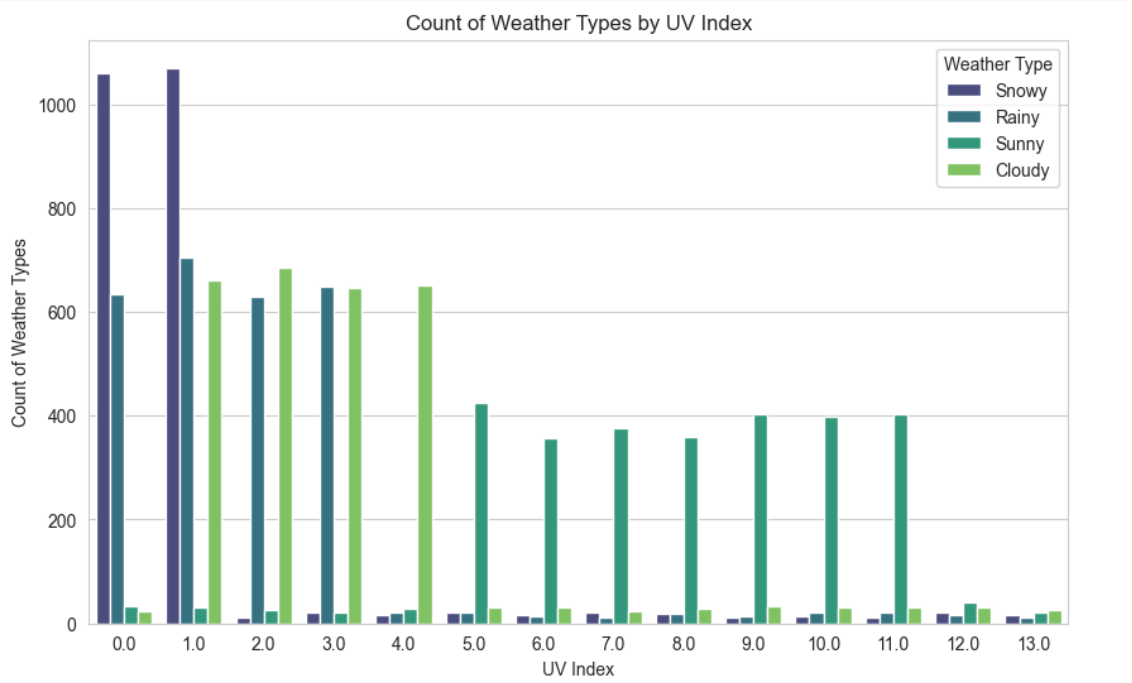
   Vyššie oblasti na grafe naznačujú vyššiu hustotu pozorovaní, zatiaľ čo nižšie oblasti ukazujú nižšiu hustotu. Pomocou tohto grafu môžete identifikovať trendy, ako sú teplejšie alebo chladnejšie obdobia v priebehu roka.



Obrázok 4: Zobrazenie grafu (kdetplot) pre teplotnú hustotu vzhľadom na ročné obdobie

3. **Počet typov počasia podľa UV indexu:** Tento graf ukazuje, ako sa mení počet jednotlivých typov počasia s rastúcim UV indexom. Môžeme si všimnúť, že s vyššími hodnotami UV indexu (napríklad nad 6) sú častejšie výskyty slnečného počasia, zatiaľ čo pri nižších hodnotách je bežnejšie daždivé počasie. Os x reprezentuje hodnoty UV indexu. Na osi y je zobrazený počet výskytov každého typu počasia pre konkrétnu hodnotu UV indexu.

   Tento počet indikuje, koľkokrát sa daný typ počasia objavil pri určitej hodnote UV indexu v množine dát.

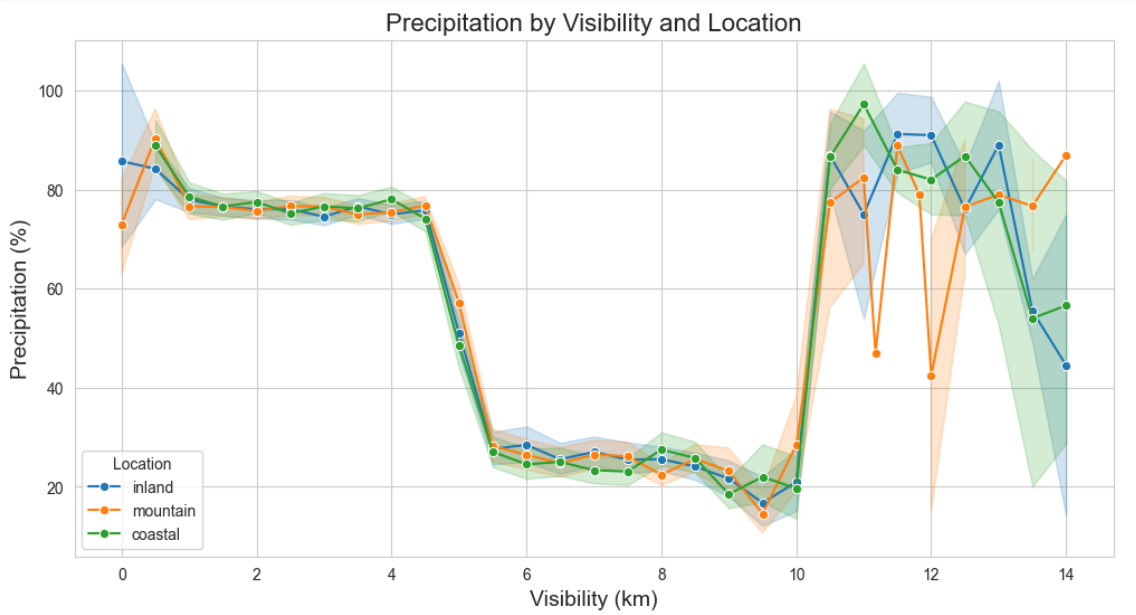


Obrázok 5: Zobrazenie grafu (countplot) pre počet typov počasia podľa UV indexu

- Sunny (slnečno): Obyčajne najvyšší počet výskytov pri vyšších hodnotách UV indexu.
- Rainy (dážď): Často sa vyskytuje pri nižších hodnotách UV indexu.
- Cloudy (oblačno) a Snowy (sneženie): Môžu mať variabilný počet výskytov v závislosti od podmienok.

4. **Zrážky podľa viditeľnosti a polohy:** Tento graf zobrazuje zrážky v závislosti od viditeľnosti a polohy. Line plot zobrazuje trend zrážok ako funkciu viditeľnosti a geografickej polohy. Os x reprezentuje viditeľnosť a os y zobrazuje percento zrážok. I Môžete vidieť, ako sa percento zrážok mení s rôznymi úrovňami viditeľnosti. Napríklad, ak je viditeľnosť vysoká, môže to naznačovať, že zrážky sú minimálne (nízke percento), zatiaľ čo pri nízkej viditeľnosti môže percento zrážok rásť v závislosti od geografickej polohy (vnútrozemská, pobrežná, horská).

   Ak máme napríklad v horách viditeľnosť 12km, pravdepodobnosť zrážok je cca 41 %.

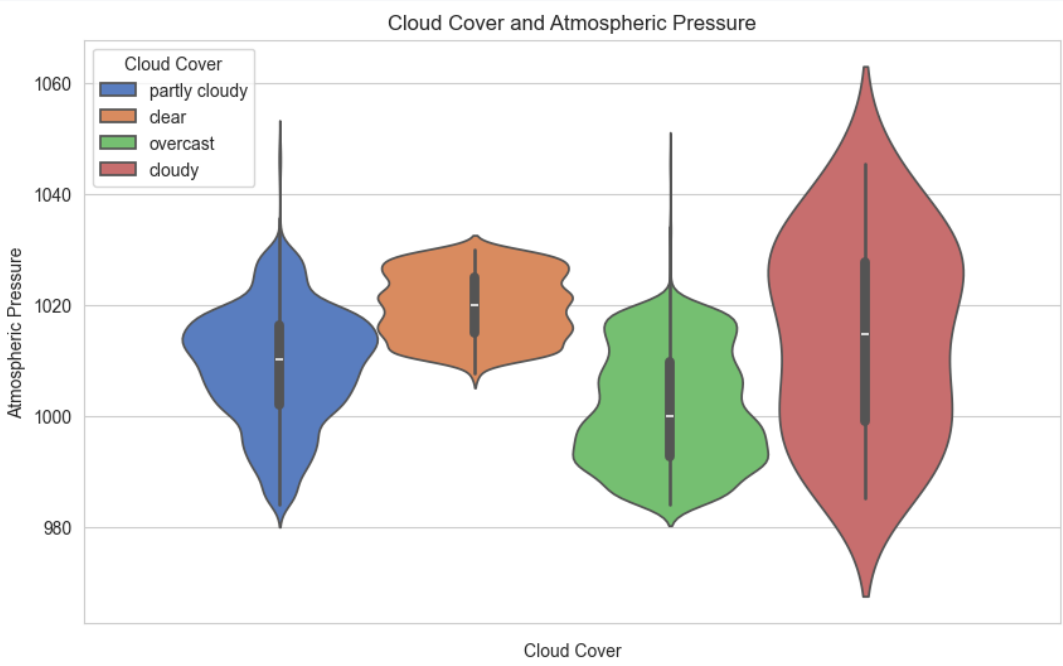


Obrázok 6: Zobrazenie grafu (lineplot) pre zrážky podľa viditeľnosti a polohy

5. **Zrážky podľa oblačnosti a atmosférického tlaku:** Tento graf zobrazuje rozloženie oblačnosti na osi x a atmosférického tlaku na osi y.

   Šírka každého tvaru v grafe zobrazuje hustotu dát v rámci kategórie oblačnosti. Čím je tvar širší, tým viac hodnôt sa v tejto oblasti nachádza. Vnútorný box plot (čierny obdĺžnik s bielou čiarou):

   - Biely bod - medián atmosférického tlaku pre danú kategóriu Cloud Cover.
   - Obdĺžnikové pole - ukazuje interkvartilový rozsah (rozpätie medzi prvým a tretím kvartilom) - kde sa nachádza 50 % dát.
   - Zvislé čiary - ukazujú rozsah dát.



Obrázok 7: Zobrazenie grafu (violinplot) pre oblačnosť a atmosférický tlak

3 Neurónová sieť

Základ tvorí funkcia *build_model()*, v ktorej je možné nastaviť počet neurónov v 1. a 2. skrytej vrstve, pričom počet prvkov vstupnej vrstvy je daný ako počet stĺpcov trénovanej množiny X (ktorá je zakódovaná a škálovaná). Ako aktivačná funkcia v skrytých vrstvách neurónu je použitá ReLU[15] a vo výstupnej vrstve je to Softmax[16]. Taktiež je tu možné nastaviť rýchlosť učenia pre Adam Optimizer[17]. V tomto prípade bude neurónová sieť používať rovnaký batch[18] - 16.

Ako funkcia straty bola použitá *sparse_categorical_crossentropy*, ktorá sa používa pri reprezentovaní údajov ako celé čísla (bol použitý Label Encoding). Pre meranie úspešnosti bola použitá *sparse categorical data*, kde sú údaje taktiež reprezentované ako celá čísla (bol použitý Label Encoding), na rozdiel od One-Hot Encoding.

3.1 Architektúra siete s pretrénovaním

Pretrénovanie nastáva vtedy, keď sa model príliš prispôsobí trénovacím údajom a nedokáže generalizovať na nové, neznáme dáta. Aby bolo dosiahnuté, bolo pridaných viac neurónov do 1. skrytej (128 neurónov) a 2. skrytej (64 neurónov) vrstvy a taktiež bol zvýšený počet epoch[19] (500). Vstupná konfigurácia siete je:

- **rýchlosť učenia:** 0.001
- **1. skrytá vrstva:** 128
- **2. skrytý vrstva:** 64
- **počet epoch:** 500

Ak sa strata na tréningových dátach neustále znižuje a dosahuje veľmi nízke hodnoty, zatiaľ čo strata na validačných dátach najskôr klesá, ale potom začne stúpať alebo oscilovať, znamená to, že model sa príliš prispôsobil trénovacím dátam a začína strácať schopnosť generalizovať na nové dáta.

[15]Aktivačná funkcia používaná v neurónových sieťach, ktorá transformuje vstupy tak, že všetky záporné hodnoty sú nahradené nulou a kladné hodnoty zostávajú nezmenené.

[16]Aktivačná funkcia, ktorá transformuje výstupy modelu na pravdepodobnosti, pričom ich súčet je 1. Pre každú triedu vypočíta pravdepodobnosť, že vstup patrí do tejto triedy.

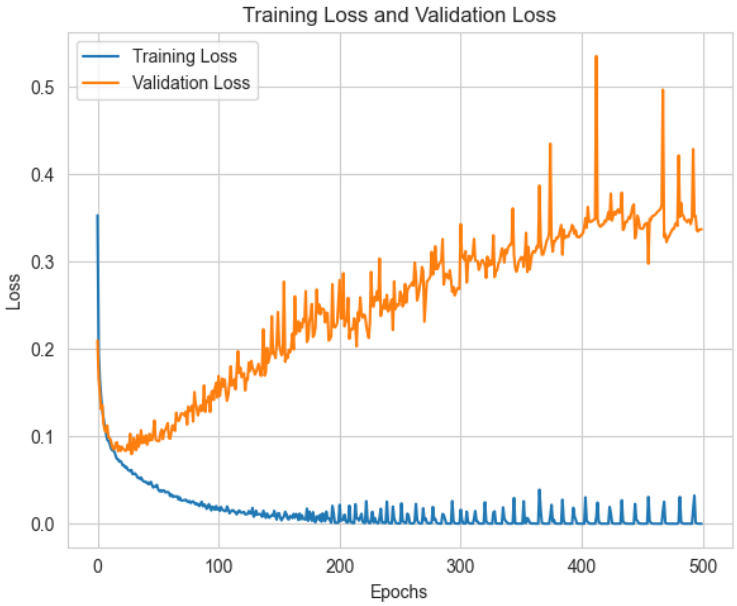
[17]adaptívne nastavuje rýchlosť učenia

[18]Počet vzoriek (riadkov) z trénovacích dát, ktoré model spracováva naraz predtým, než aktualizuje váhy

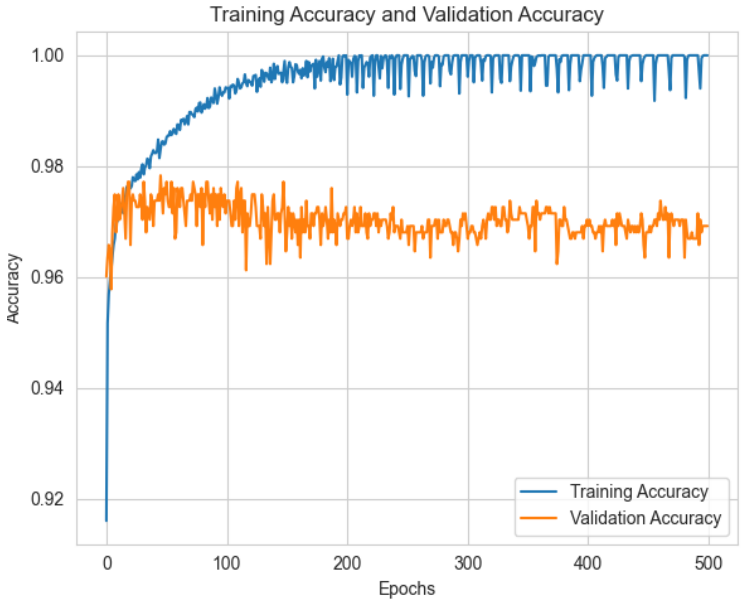
[19]Jeden kompletný prechod všetkými trénovacími dátam

Ak presnosť na tréningových dátach neustále stúpa a dosahuje veľmi vysoké hodnoty (blíži sa k 1), zatiaľ čo presnosť na validačných dátach dosiahne určitý bod a potom začne klesať alebo stagnovať, znamená to, že model sa príliš učí detaily trénovacích dát (jeho schopnosť generalizovať sa zhoršuje).

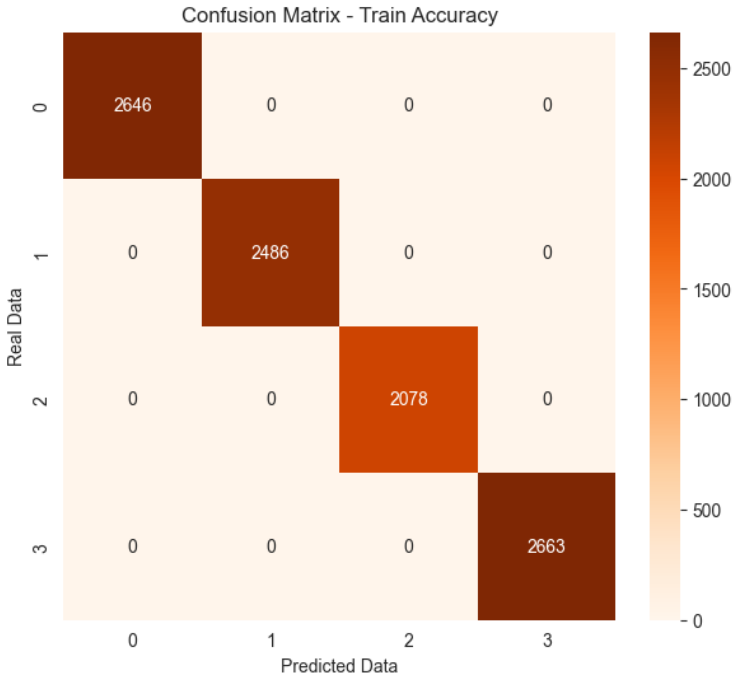
V tomto prípade dosiahla trénovacia úspešnosť hodnotu **1** a testovacia úspešnosť hodnotu **0.977272**.



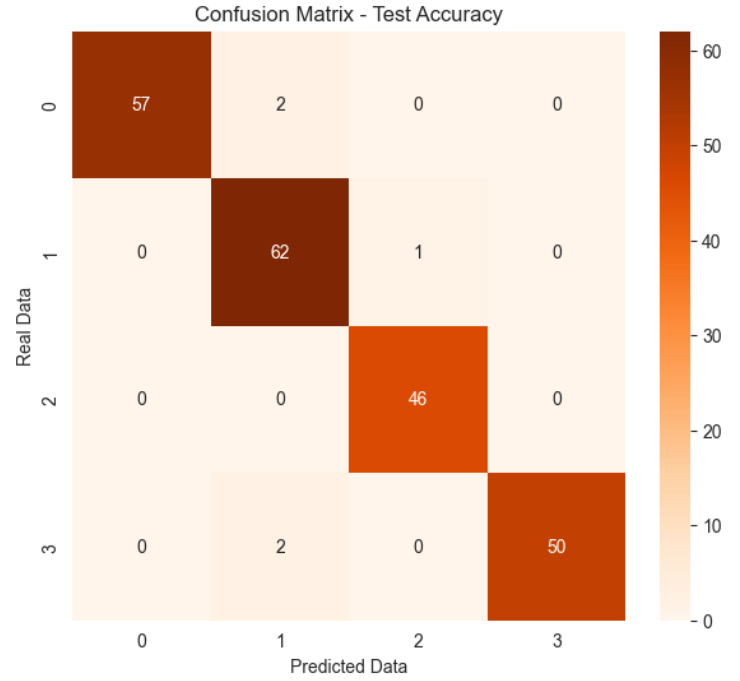
Obrázok 8: Graf trénovacej a validačnej straty



Obrázok 9: Graf trénovacej a validačnej úspešnosti



Obrázok 10: Zobrazenie konfúznej matice pre trénovaciu úspešnosť



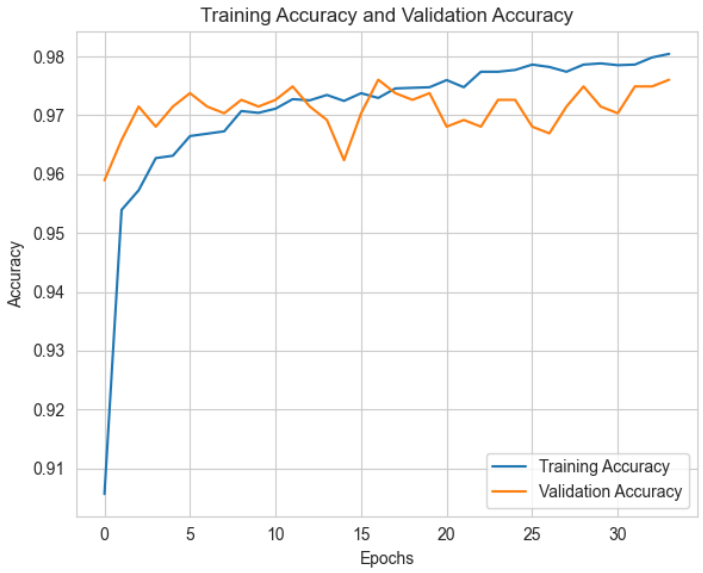
Obrázok 11: Zobrazenie konfúznej matice pre testovaciu úspešnosť

3.2 Architektúra siete bez pretrénovania

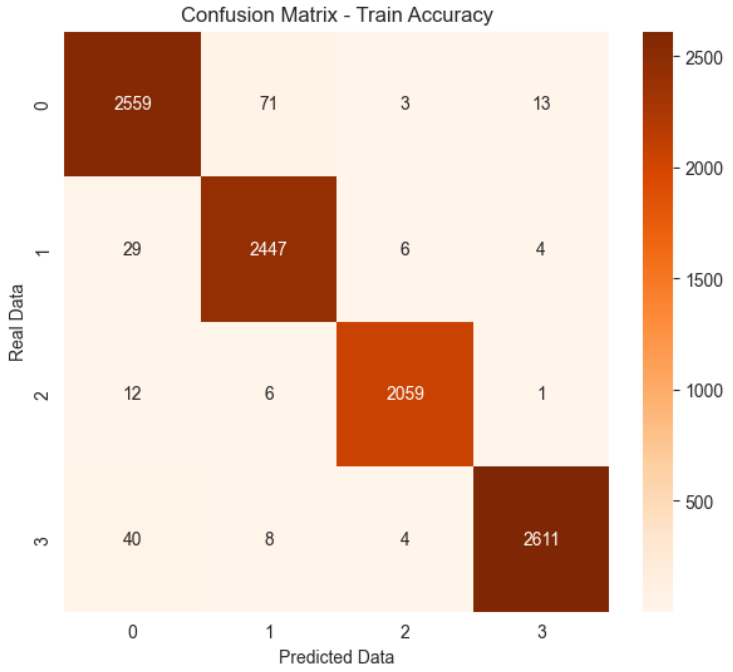
V tomto prípade pracujeme s konfiguráciou uvedenou v sekcií 3.1, avšak aby sme odstránili pretrénovanie, pridáme **Early Stopping**, pre skoré zastavenie trénovania v prípade, že sa vyskytne náznak pretrénovania. To môže ovplyvniť počet epôch, ktorý sa v tomto prípade zníži (ako je možné vidieť na grafoch).

Ako je možné na grafoch vidieť, táto neurónová sieť nemá vďaka použitiu Early Stoppingu vyššie uvedené známky pretrénovania.

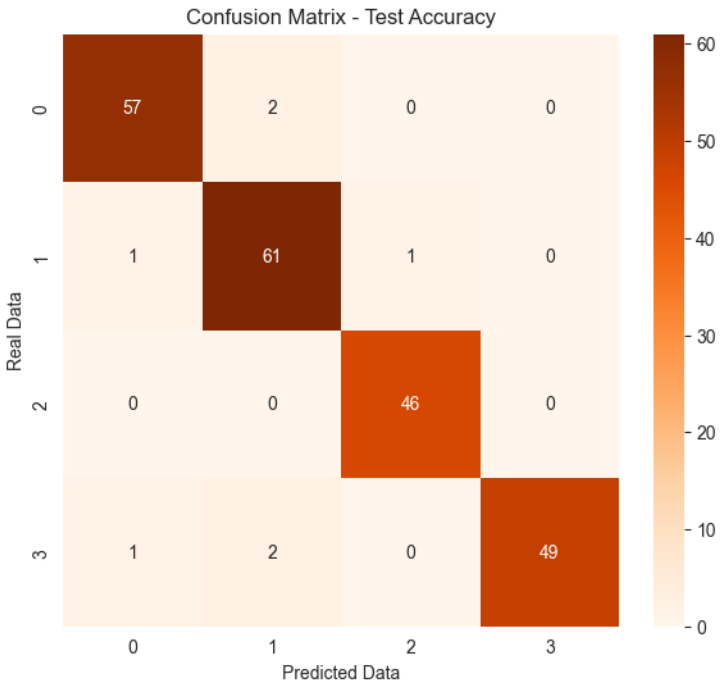
Obrázok 12: Graf trénovacej a validačnej straty s použitím Early Stopping



Obrázok 13: Graf trénovacej a validačnej úspešnosti s použitím Early Stopping



Obrázok 14: Zobrazenie konfúznej matice pre trénovaciu úspešnosť s použitím Early Stopping



Obrázok 15: Zobrazenie konfúznej matice pre testovaciu úspešnosť s použitím Early Stopping

V tomto prípade dosiahla trénovacia úspešnosť hodnotu **0.98004** a testovacia úspešnosť hodnotu **0.968181**, čo indikuje zlepšenie, oproti predchádzajúcej sieti (bez Early Stoppingu).

3.3 Experimenty s trénovaním siete

Pre zobrazenie rozdielov v trénovaniach a výsledných úspešnostiach na základe rôznych vstupných hyperparametrov bolo vykonaných 5 experimentov, pričom v prvých troch experimentoch bola zmenená hodnota parametra rýchlosti učenia a v ďalších dvoch bol zmenený počet neurónov v 1. a 2. skrytej vrstve.

Všetky tieto zmeny, vrátane výpisu úspešnosti jednotlivých experimentov sú zapísané v tabuľke nižšie.

Číslo	Rýchlosť učenia	1. vrstva	2. vrstva	Trénovacia úspešnosť	Testovacia úspešnosť
1	0.01	64	32	0.996354	0.981818
2	0.0001	64	32	0.976299	0.972727
3	0.01	64	32	0.986934	0.972727
4	0.01	128	16	0.980958	0.968182
5	0.01	32	32	0.980654	0.968182

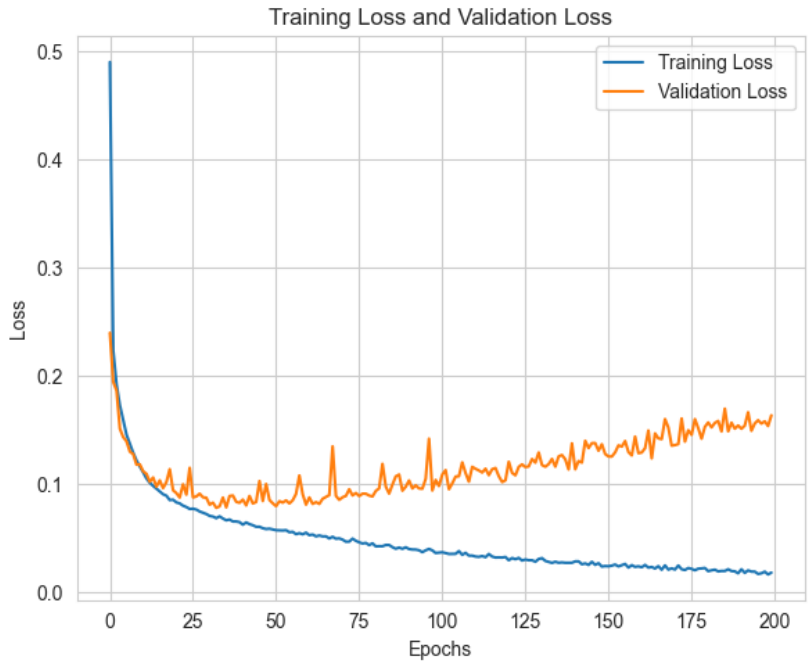
Tabuľka 3: Konfigurácia jednotlivých experimentov a presnosť trénovania a testovania

Výber najlepšieho a najhoršieho testu T bol zvolený na základe priemeru trénovacej t_1 a testovacej t_2 úspešnosti:

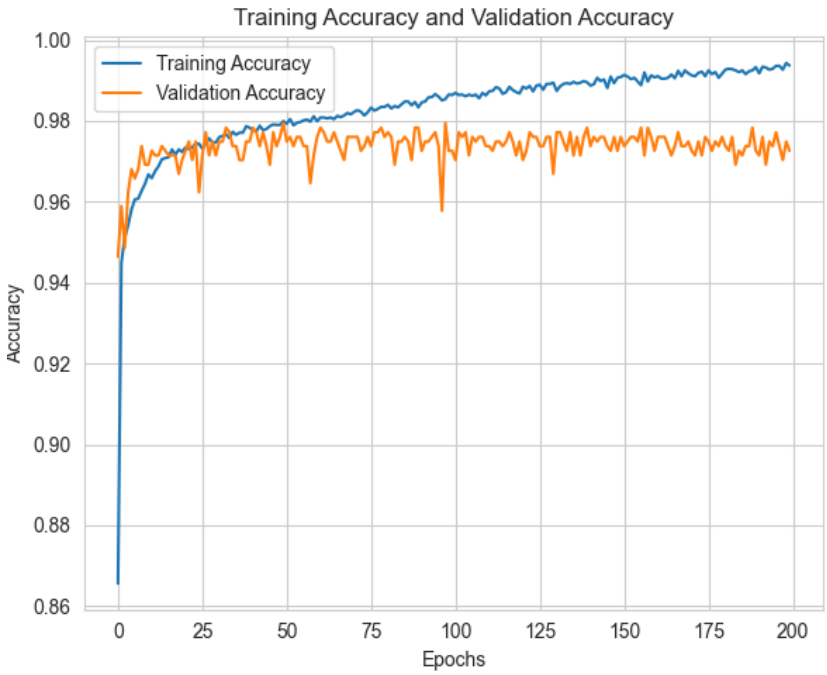
$$T = \frac{t_1 + t_2}{2},$$

pričom tento priemer bol vykonaný pre každý experiment v tabuľke. Najvyššia hodnota T predstavuje najlepší experiment (č. 1 s priemerom 0.989085) a najmenšia hodnota predstavuje najhorší (č. 5 s priemerom 0.974418) experiment.

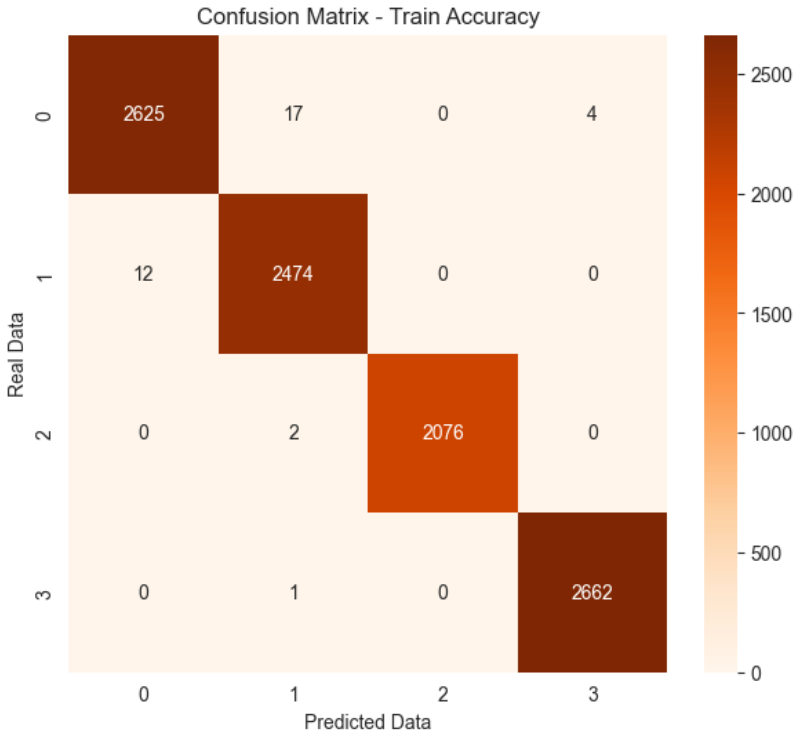
- **Grafy úspešnosti a strát a konfúzne matice pre najlepší experiment**



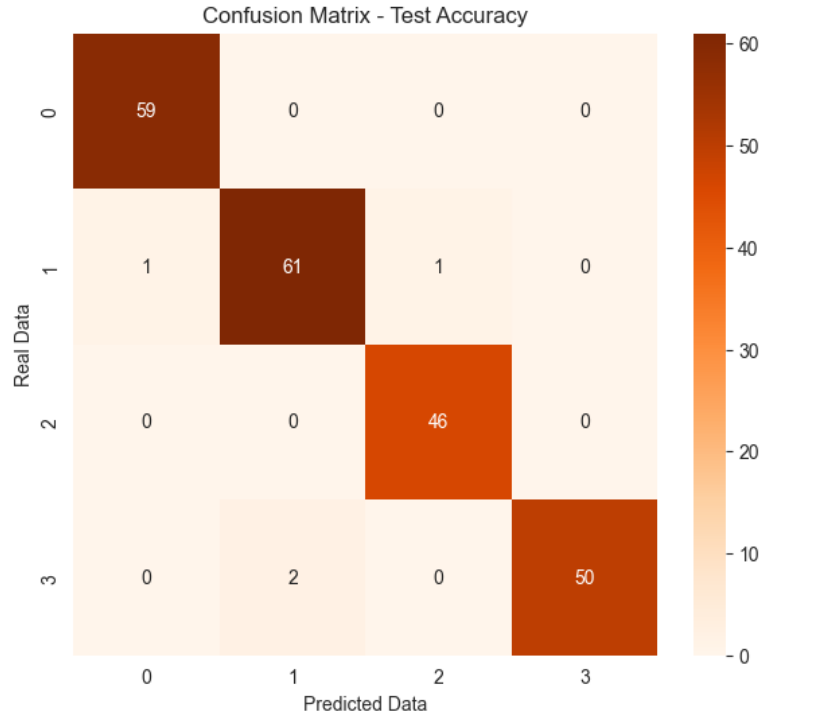
Obrázok 16: Graf trénovacej a validačnej straty pre najlepší experiment



Obrázok 17: Graf trénovacej a validačnej úspešnosti pre najlepší experiment

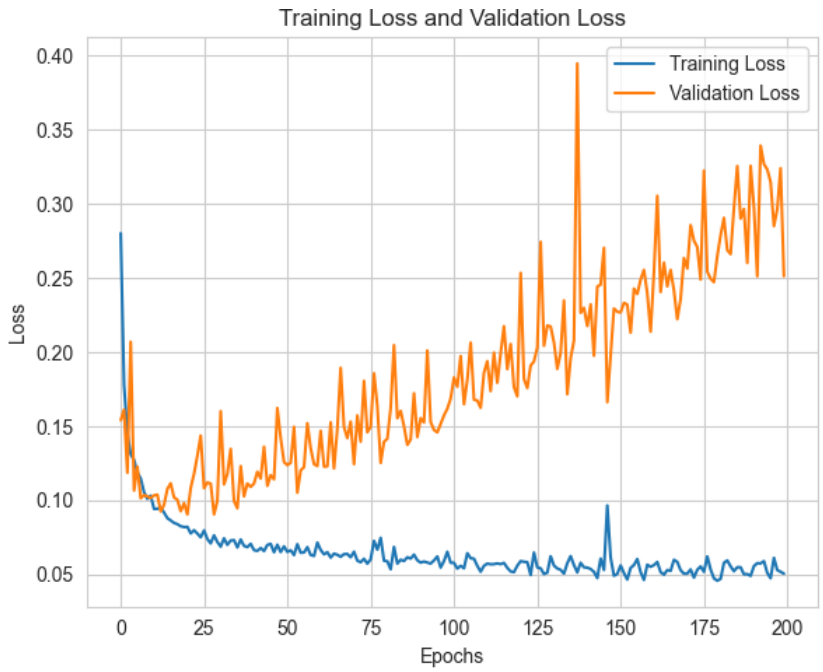


Obrázok 18: Konfúzna matica trénovacej množiny pre najlepší experiment

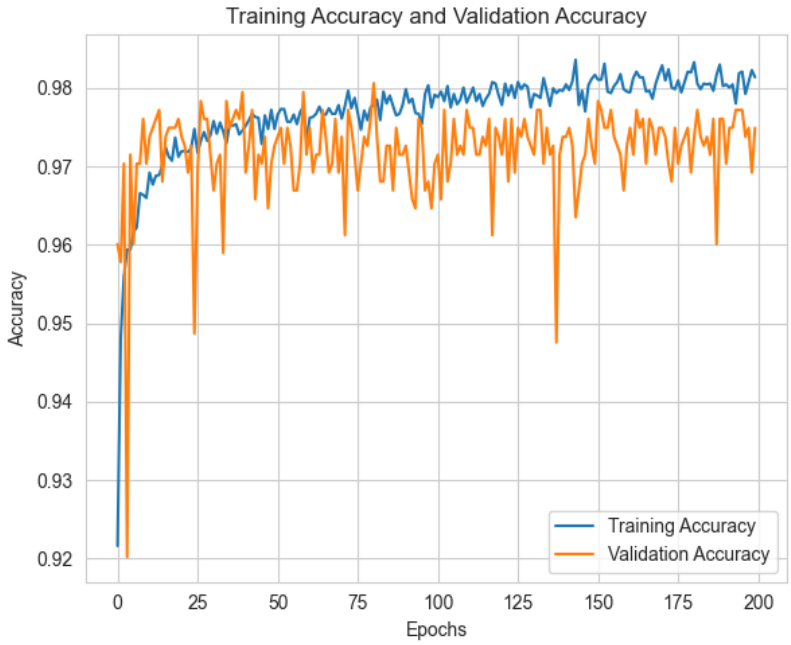


Obrázok 19: Konfúzna matica testovacej množiny pre najlepší experiment

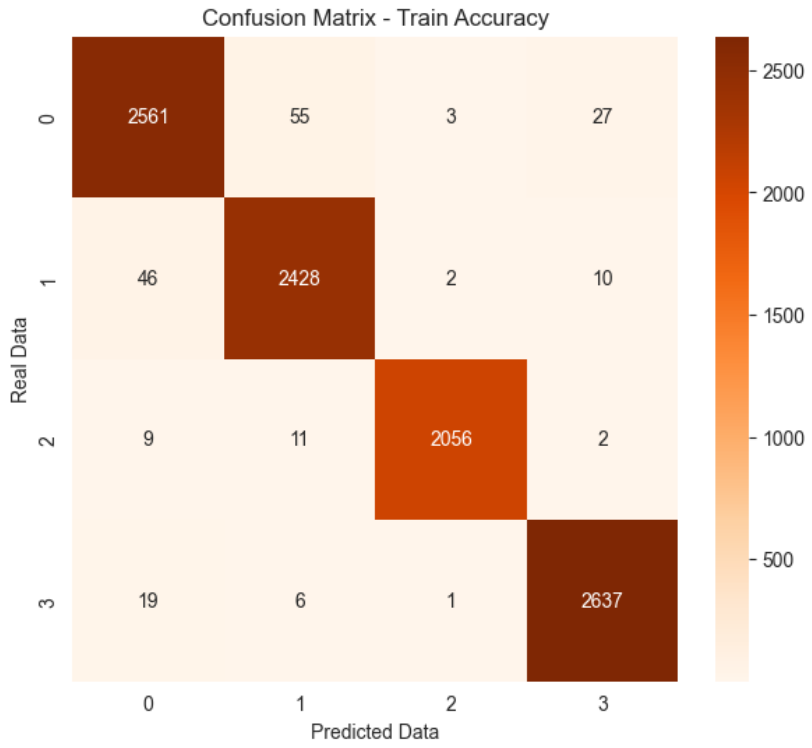
- **Grafy úspešnosti a strát a konfúzne matice pre najhorší experiment**



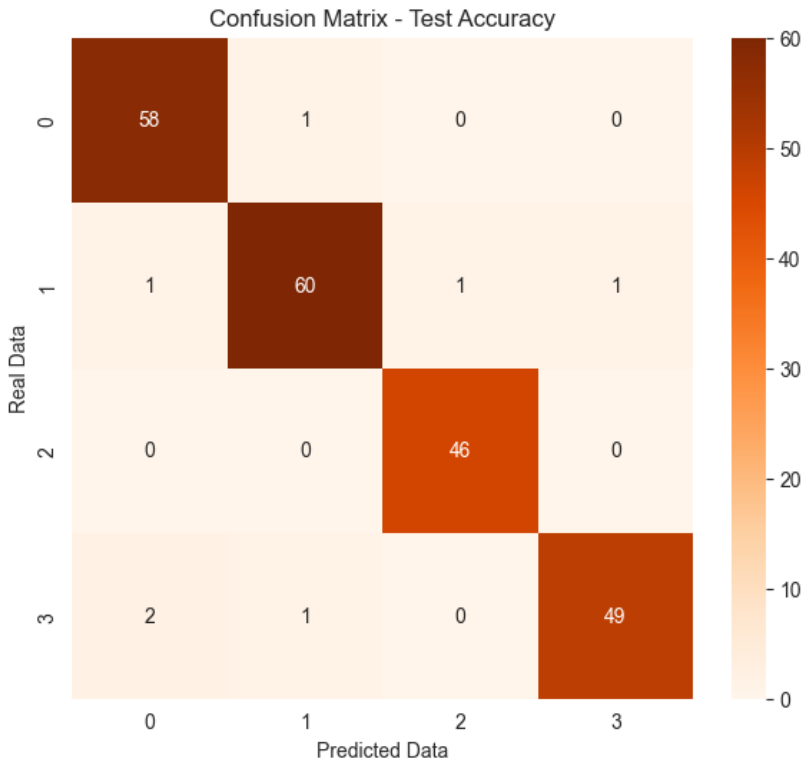
Obrázok 20: Graf trénovacej a validačnej straty pre najhorší experiment



Obrázok 21: Graf trénovacej a validačnej úspešnosti pre najhorší experiment



Obrázok 22: Konfúzna matica trénovacej množiny pre najhorší experiment

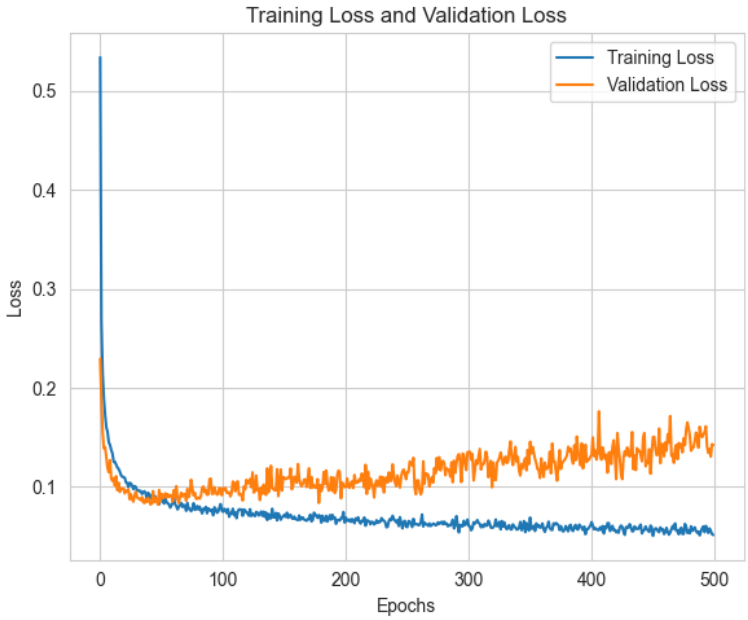


Obrázok 23: Konfúzna matica testovacej množiny pre najhorší experiment

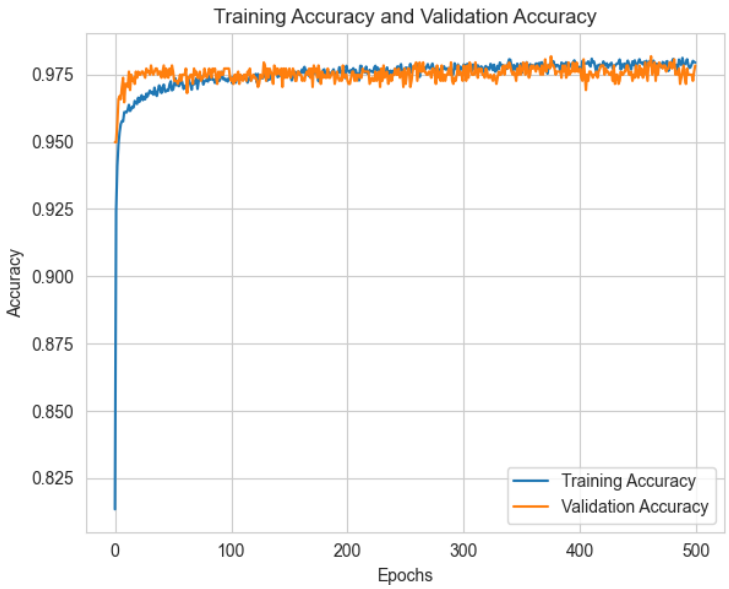
3.3.1 Dropout vrstvy

Dropout vrstvy môžu pomôcť znížiť pretrénovanie pridaním náhodného vyraďovania neurónov počas trénovania. Túto vsrtvu pridávame vždy s pravdepodobnosťou vypnutia neurónov (napríklad 0.2, resp. 20 % pravdepodobnosť, že sa neuróny vypnú.

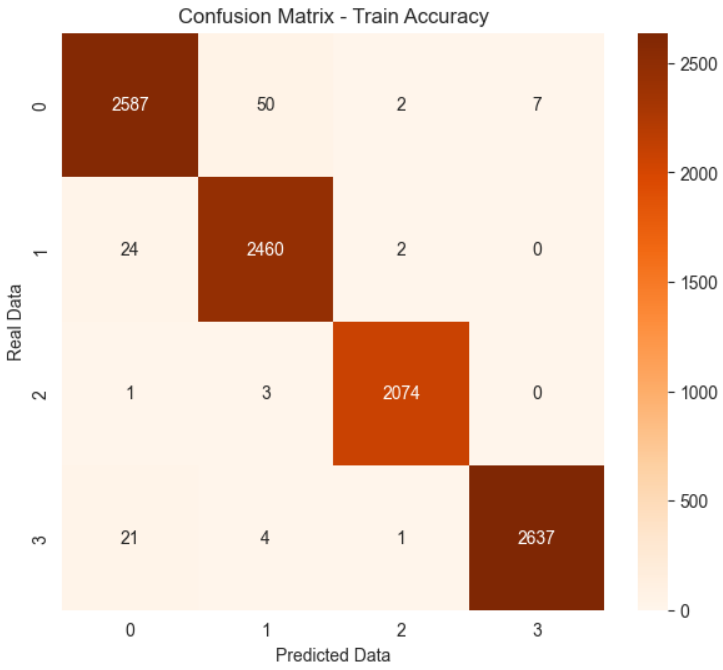
Dropout vrstvu môžeme pridať do každej skrytej vrstvy a tak povypínať neuróny s rôznou pravdepodobnosťou. V tomto prípade pracujeme s konfiguráciou uvedenou v sekcií 3.1, pričom boli pridané 2 dropout vrstvy do skrytej vrstvy neurónovej siete - obidve s pravdepodobnosťami 40%.



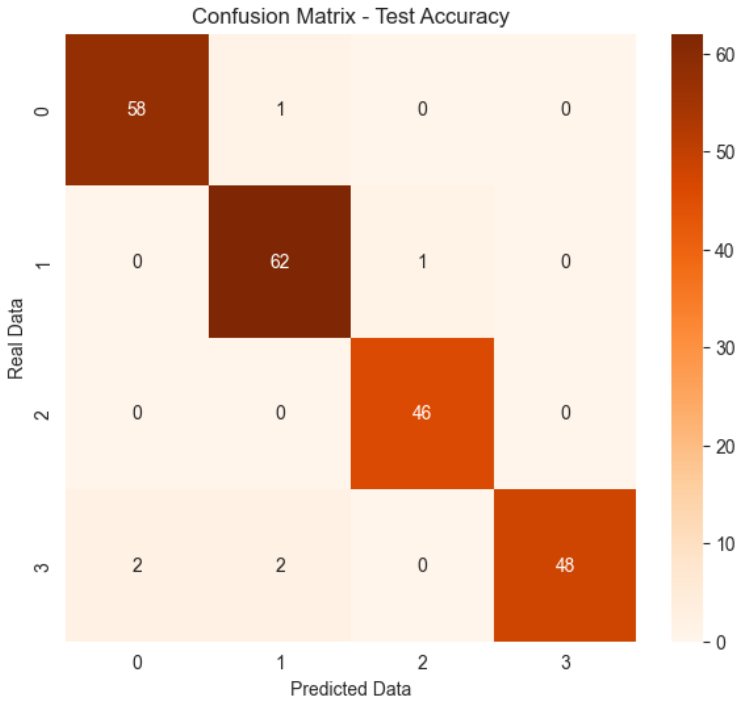
Obrázok 24: Graf trénovacej a validačnej straty pre neurónovú sieť s droupout vrstvami



Obrázok 25: Graf trénovacej a validačnej úspešnosti pre neurónovú sieť s droupout vrstvami



Obrázok 26: Zobrazenie konfúznej matice pre trénovaciu úspešnosť pre neurónovú sieť s droupout vrstvami



Obrázok 27: Zobrazenie konfúznej matice pre testovaciu úspešnosť pre neurónovú sieť s droupout vrstvami

Uvedené parametre neurónovej siete boli vybrané zámerne - aby bolo možné demonštrovať na rovnakej pretrénovaj sieti rôzne úspešnosti. Ako je možné vidieť, úspešnosť sa oproti pôvodnej sieti zlepšila (trénovacia úspešnosť má hodnotu 0.98835 a testovacia úspešnosť má hodnotu 0.972727).

Záver

Cieľom tohto zadania bolo kategorizovať počasia do štyroch kategórií (Rainy, Cloudy, Sunny, Snowy), pričom ako vstupné parameter boli použité množiny Cloud Cover, Season, Location, Temperature, Humidity, Wind Speed, Precipitation (%), Atmospheric Pressure, UV Index, Visibility (km).

Dáta boli očistené od outlierov, chýbajúcich hodnôt alebo odchýlok, boli zakódované Label Encoderom a následne normalizované MinMaxScalerom. S ďalej pracovali pri trénovaní neurónovej siete, pričom bolo vykonaných množstvo experimentov s cieľom zistiť, ako ovplyvňuje úspešnosť siete zmena jednotlivých hyperparametrov.

Taktiež bol pomocou Exploratory Data Analysis (EDA) analyzovaný dataset pred zakódovaním a následne bolo (aj) pomocou korelačnej matice nájdených 5 zaujímavých vzťahov, pre ktoré boli vykreslené a popísané príslušné grafy.