SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

ZADANIE 1 : UMELÉ NEURÓNOVÉ SIETE
SEMINÁRNA PRÁCA

2021 Ladislav Rajcsányi

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

ZADANIE 1 : UMELÉ NEURÓNOVÉ SIETE
SEMINÁRNA PRÁCA

Študijný program:	Aplikovaná informatika
Predmet:	I-SUNS – Strojové učenie a neurónové siete
Prednášajúci:	prof. Dr. Ing. Miloš Oravec
Cvičiaci:	Ing. Zuzana Bukovčiková
	Ing. Vanesa Andicsová
	Ing. Dominik Sopiak, PhD.

Bratislava 2021 **Ladislav Rajcsányi**

Obsah

Zoznam obrázkov a tabuliek

Zoznam skratiek

API Application Programming Interface

ML Machine Learning

NaN Not a Number

NS Neurónová Sieť

Úvod

Hlavným cieľom tohto zadania je predpovedanie kvality vína pomocou (Umelých) Neurónových Sietí - NS (Artifical Neural Network). Na vypracovanie zadania sme použili poskytnuté dátové súbory, ktoré sú rozdelené na trénovacie a testovacie dáta. Trénovacie dáta použijeme na natrénovanie neurónových sietí a v prípade potreby môžeme z nich vybrať aj validačné dáta, ktoré môžeme používať na monitorovanie úspešnosti na predtým nevidených dát počas fázy trénovania. Z nižšie popísaných dát nám vyplýva, že táto úloha je triediaca (klasifikačná).

Dáta sú uložené v CSV súboroch. Trénovacie (wine_train.csv) aj testovacie (wine_test.csv) dáta majú 13 stĺpcov, pričom jednotlivé stĺpce reprezentujú jednotlivé vlastnosti vín, ktoré sú nasledovné:

1. Stála kyslosť (angl.: fixed acidity) : Väčšina kyselín vo víne je stála alebo neprchavá (neodparujú sa ľahko) [kyselina vínna - g / dm^3].

2. Prchavá kyslosť (angl.: volatile acidity) : Množstvo kyseliny octovej vo víne, ktorá pri príliš vysokých hodnotách môže viesť k nepríjemnej octovej chuti [kyselina octová - g / dm^3].

3. Kyselina citrónová (angl.: citric acid) : Kyselina citrónová, ktorá sa nachádza v malých množstvách, môže dodať sviežosť a chuť [g / dm^3].

4. Zvyškový cukor (angl.: residual sugar) : Množstvo cukru, ktorý zostane po ukončení kvasenia, je zriedkavé. Vína s obsahom menej ako 1 gram/liter a vína s obsahom viac ako 45 gramov/liter sa považujú za sladké [g / dm^3].

5. Chloridy (angl.: chlorides) : Množstvo soli vo víne [chlorid sodný - g / dm^3].

6. Voľný oxid siričitý (angl.: free sulfur dioxide) : Voľná forma SO2 existuje v rovnováhe medzi molekulárnym SO2 (ako rozpustený plyn) a disiričitanovým iónom; bráni mikrobiálnemu rastu a oxidácii vína [mg / dm^3].

7. Celkový oxid siričitý (angl.: total sulfur dioxide) : Množstvo voľnej a viazanej formy S02; v nízkych koncentráciách, SO2 sa vo víne väčšinou nedá zistiť, ale pri koncentráciách voľného SO2 nad 50 ppm, SO2 sa stáva cítiteľným v chuti vína. [mg / dm^3].

8. Hustota (angl.: density) : Hustota vody je blízka hustote vody v závislosti od percenta alkoholu a obsahu cukru [g / cm^3].

9. pH : Opisuje, ako kyslé alebo zásadité je víno na stupnici od 0 (veľmi kyslé) do 14 (veľmi zásadité); väčšina vín sa pohybuje v rozmedzí 3 - 4 na stupnici pH.

10. sírany (angl.: sulphates) : Prídavná látka do vína, ktorá môže prispievať k zvýšeniu hladiny plynného oxidu siričitého (SO2), ktorý pôsobí antimikrobiálne a antioxidačne [síran draselný - g/dm^3].

11. alkohol (angl.: alcohol) : Percentuálny obsah alkoholu vo víne [objemové %].

12. typ (angl.: type) : Druh vína (0 - biele, 1 - červené)

13. **kvalita** (angl.: quality) : Výstupná premenná (na základe senzorických údajov), 0 - nízka kvalita, 1 - vysoká kvalita

1 Použité technológie

1.1 Pandas

Pandas je rýchly, výkonný, flexibilný a ľahko použiteľný open source nástroj na analýzu a manipuláciu s údajmi, postavený na programovacom jazyku Python [1].

1.2 Matplotlib

Matplotlib je komplexná knižnica na vytváranie statických, animovaných a interaktívnych vizualizácií v jazyku Python [2].

1.3 Seaborn

Seaborn je knižnica na vizualizáciu údajov v jazyku Python založená na matplotlib. Poskytuje vysokoúrovňové rozhranie na kreslenie atraktívnej a informatívnej štatistickej grafiky [3].

1.4 Scikit-Learn

- Jednoduché a efektívne nástroje na prediktívnu analýzu údajov
- Prístupné pre každého a opakovane použiteľné v rôznych kontextoch
- Postavené na NumPy, SciPy a matplotlib
- Open Source, komerčne použiteľný - licencia BSD [4]

1.5 TensorFlow

TensorFlow je komplexná open source platforma pre strojové učenie (angl.: Machine Learning - ML). Má komplexný, flexibilný ekosystém nástrojov, knižníc a komunitných zdrojov, ktorý umožňuje výskumníkom posúvať najnovšie poznatky v oblasti ML a vývojárom ľahko vytvárať a nasadzovať aplikácie využívajúce ML [5].

1.6 Keras

Keras je API určené pre ľudí, nie pre stroje. Keras sa riadi osvedčenými postupmi na zníženie kognitívnej záťaže: ponúka konzistentné a jednoduché API, minimalizuje počet činností používateľa potrebných pre bežné prípady použitia a poskytuje jasné a použiteľné chybové hlásenia. Má tiež rozsiahlu dokumentáciu a príručky pre vývojárov [6].

2 Implementácia

2.1 Chýbajúce Údaje (Missing Data)

Riešenie tohto zadania si začneme načítaním vopred nachystaných trénovacích aj testovacích dát pomocou metódy **read_csv** [7], ktorý nám ukladá načítané dáta do Pandas DataFrame-u. Po načítaní si môžeme pozrieť dáta pomocou metódy **head** [8], ktorá nám vráti nasledujúcu tabuľku :

type	fixed acidity	volatile acidity	citric acid	citric acid	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	5.9	0.18	0.29	4.6	0.032	68.0	137.0	0.99159	3.21	0.38	11.3	1
0	5.8	0.18	0.28	1.3	0.034	9.0	94.0	0.99092	3.21	0.52	11.2	1
0	6.0	0.495	0.27	5.0	0.157	17.0	129.0	0.99396	3.03	0.36	9.3	0
1	6.4	0.79	0.04	2.2	0.061	11.0	17.0	0.99588	3.53	0.65	10.4	1
1	10.8	0.47	0.43	2.1	0.171	27.0	66.0	0.9982	3.17	0.76	10.8	1
0	7.6	0.13	0.34	9.3	0.062	40.0	126.0	0.9966	3.21	0.39	9.6	0
1	8.3	0.65	0.1	2.9	0.089	17.0	40.0	0.99803	3.29	0.55	9.5	0
0	8.2	0.37	0.36	1.0	0.034	17.0	93.0	0.9906	3.04	0.32	11.7	1
1	5.0	0.38	0.01	1.6	0.048	26.0	60.0	0.99084	3.7	0.75	14.0	1
0	7.4	0.19	0.3	12.8	0.053	48.5	229.0	0.9986	3.14	0.49	9.1	1

Tabuľka 2: Výstup metódy **head** na trénovacích dátach

Následne si môžeme zobraziť jednoduché informácie o trénovacích a testovacích dátach pomocou metódy **info** [9]. Táto metóda nám vráti nasledujúce výsledky:

```
Information about the Training Data Set

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5197 entries, 0 to 5196
Data columns (total 13 columns):
 #   Column              Non-Null Count Dtype
---  ------              -------------- -----
 0   type                5197 non-null  int64
 1   fixed acidity       5188 non-null  float64
 2   volatile acidity    5192 non-null  float64
 3   citric acid         5195 non-null  float64
 4   residual sugar      5195 non-null  float64
 5   chlorides           5195 non-null  float64
 6   free sulfur dioxide 5197 non-null  float64
 7   total sulfur dioxide 5197 non-null float64
 8   density             5197 non-null  float64
 9   pH                  5188 non-null  float64
 10  sulphates           5194 non-null  float64
 11  alcohol             5197 non-null  float64
 12  quality             5197 non-null  int64
dtypes: float64(11), int64(2)
memory usage: 527.9 KB
None
```

```
Information about the Testing Data Set

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1300 entries, 0 to 1299
Data columns (total 13 columns):
 #   Column               Non-Null Count Dtype
---  ------               -------------- -----
 0   type                 1300 non-null  int64
 1   fixed acidity        1299 non-null  float64
 2   volatile acidity     1297 non-null  float64
 3   citric acid          1299 non-null  float64
 4   residual sugar       1300 non-null  float64
 5   chlorides            1300 non-null  float64
 6   free sulfur dioxide  1300 non-null  float64
 7   total sulfur dioxide 1300 non-null  float64
 8   density              1300 non-null  float64
 9   pH                   1300 non-null  float64
 10  sulphates            1299 non-null  float64
 11  alcohol              1300 non-null  float64
 12  quality              1300 non-null  int64
dtypes: float64(11), int64(2)
memory usage: 132.2 KB
None
```

Z týchto údajov vidíme, že niektoré stĺpce/riadky obsahujú prázdne hodnoty tzv. NaN alebo Null Value. Môžeme vypísať aj to, že aké percento tvoria NaN hodnoty v jednotlivých stĺpcoch. Jednoduchý príkaz nám vráti nasledujúce údaje:

```
Number of NaN Values in the Training Data Set before dealing with NaN Values

Column type has 0 (0.0 %) NaN value(s)
Column fixed acidity has 9 (0.173 %) NaN value(s)
Column volatile acidity has 5 (0.096 %) NaN value(s)
Column citric acid has 2 (0.038 %) NaN value(s)
Column residual sugar has 2 (0.038 %) NaN value(s)
Column chlorides has 2 (0.038 %) NaN value(s)
Column free sulfur dioxide has 0 (0.0 %) NaN value(s)
Column total sulfur dioxide has 0 (0.0 %) NaN value(s)
Column density has 0 (0.0 %) NaN value(s)
Column pH has 9 (0.173 %) NaN value(s)
Column sulphates has 3 (0.058 %) NaN value(s)
Column alcohol has 0 (0.0 %) NaN value(s)
Column quality has 0 (0.0 %) NaN value(s)
```

```
Number of NaN Values in the Testing Data Set before dealing with NaN Values

Column type has 0 (0.0 %) NaN value(s)
Column fixed acidity has 1 (0.077 %) NaN value(s)
Column volatile acidity has 3 (0.231 %) NaN value(s)
Column citric acid has 1 (0.077 %) NaN value(s)
Column residual sugar has 0 (0.0 %) NaN value(s)
Column chlorides has 0 (0.0 %) NaN value(s)
Column free sulfur dioxide has 0 (0.0 %) NaN value(s)
Column total sulfur dioxide has 0 (0.0 %) NaN value(s)
Column density has 0 (0.0 %) NaN value(s)
Column pH has 0 (0.0 %) NaN value(s)
Column sulphates has 1 (0.077 %) NaN value(s)
Column alcohol has 0 (0.0 %) NaN value(s)
Column quality has 0 (0.0 %) NaN value(s)
```

Teraz nastáva otázka, čo by sme mali robiť s chýbajúcimi dátami. Existujú rôzne stratégie pre túto situáciu. Môžeme jednoducho vymazať tie vzorky (riadky), ktoré obsahujú NaN hodnotu. Druhá možnosť je nahradiť nejakou hodnotou. V tomto zadaní program umožní používateľovi si vybrať stratégiu pre túto situáciu. Štandardne program používa nahraďovaciu stratégiu, pričom NaN hodnoty nahradíme priemernou hodnotou daného príznaku (stĺpce). Pre testovacie dáta program automaticky nahradí chýbajúce údaje priemernou hodnotou, totiž nie je odporúčané vymazávanie vzoriek. Tiež podporuje, ak používateľ chce vymazať vzorky z trénovacích dát stačí mu napísať písmeno ‘d‘, keď uvidí nasledujúci výpis.

```
Please select strategy (D)rop or (F)ill Nan Values: (D/[F])
```

V tomto dokumente pre jednoduchosť budeme používať nahraďovaciu stratégiu. Po nahradení chýbajúcich údajov môžeme zobraziť štatistiku ešte raz a uvidíme, že už nemáme žiadne chýbajúce údaje v dátach. Podobne ako chýbajúce dáta aj duplikáty nám môžu znižovať úspešnosť modelu, tým pádom, je odporúčané ich čím skôr vyhodiť.

```
Number of NaN Values in the Training Data Set after filling NaN Values

Column type has 0 (0.0 %) NaN value(s)
Column fixed acidity has 0 (0.0 %) NaN value(s)
Column volatile acidity has 0 (0.0 %) NaN value(s)
Column citric acid has 0 (0.0 %) NaN value(s)
Column residual sugar has 0 (0.0 %) NaN value(s)
Column chlorides has 0 (0.0 %) NaN value(s)
Column free sulfur dioxide has 0 (0.0 %) NaN value(s)
Column total sulfur dioxide has 0 (0.0 %) NaN value(s)
Column density has 0 (0.0 %) NaN value(s)
Column pH has 0 (0.0 %) NaN value(s)
Column sulphates has 0 (0.0 %) NaN value(s)
Column alcohol has 0 (0.0 %) NaN value(s)
Column quality has 0 (0.0 %) NaN value(s)

Number of NaN Values in the Testing Data Set after filling NaN Values

Column type has 0 (0.0 %) NaN value(s)
Column fixed acidity has 0 (0.0 %) NaN value(s)
Column volatile acidity has 0 (0.0 %) NaN value(s)
Column citric acid has 0 (0.0 %) NaN value(s)
Column residual sugar has 0 (0.0 %) NaN value(s)
Column chlorides has 0 (0.0 %) NaN value(s)
Column free sulfur dioxide has 0 (0.0 %) NaN value(s)
Column total sulfur dioxide has 0 (0.0 %) NaN value(s)
Column density has 0 (0.0 %) NaN value(s)
Column pH has 0 (0.0 %) NaN value(s)
Column sulphates has 0 (0.0 %) NaN value(s)
Column alcohol has 0 (0.0 %) NaN value(s)
Column quality has 0 (0.0 %) NaN value(s)
```

```
Information about the Training Data Set after filling NaN Values and dropped duplicates

<class 'pandas.core.frame.DataFrame'>
Int64Index: 4424 entries, 0 to 5196
Data columns (total 13 columns):
 #   Column               Non-Null Count Dtype
---  ------               -------------- -----
 0   type                 4424 non-null  int64
 1   fixed acidity        4424 non-null  float64
 2   volatile acidity     4424 non-null  float64
 3   citric acid          4424 non-null  float64
 4   residual sugar       4424 non-null  float64
 5   chlorides            4424 non-null  float64
 6   free sulfur dioxide  4424 non-null  float64
 7   total sulfur dioxide 4424 non-null  float64
 8   density              4424 non-null  float64
 9   pH                   4424 non-null  float64
 10  sulphates            4424 non-null  float64
 11  alcohol              4424 non-null  float64
 12  quality              4424 non-null  int64
dtypes: float64(11), int64(2)
memory usage: 483.9 KB
None

 Information about the Testing Data Set after filling NaN Values and dropped duplicates

 <class 'pandas.core.frame.DataFrame'>
 Int64Index: 1238 entries, 0 to 1299
 Data columns (total 13 columns):
  #   Column               Non-Null Count Dtype
 ---  ------               -------------- -----
  0   type                 1238 non-null  int64
  1   fixed acidity        1238 non-null  float64
  2   volatile acidity     1238 non-null  float64
  3   citric acid          1238 non-null  float64
  4   residual sugar       1238 non-null  float64
  5   chlorides            1238 non-null  float64
  6   free sulfur dioxide  1238 non-null  float64
  7   total sulfur dioxide 1238 non-null  float64
  8   density              1238 non-null  float64
  9   pH                   1238 non-null  float64
  10  sulphates            1238 non-null  float64
  11  alcohol              1238 non-null  float64
  12  quality              1238 non-null  int64
 dtypes: float64(11), int64(2)
 memory usage: 135.4 KB
 None
```

Keďže sme už vyčistili dáta, môžeme prejsť na Prieskumnú Analýzu Údajov.

2.2 Prieskumná Analýza Údajov (Exploratory Data Analysis)

V tejto sekcií si prejdeme niektoré vlastnosti trénovacích aj testovacích dát. Hneď môžeme začať so zobrazením popisu dát. Popis dát môžeme vygenerovať pomocou metódy **describe** [10]. Tento popis obsahuje informácie o jednotlivých stĺpcoch, hlavne čo sa týka minimálnej/maximálnej hodnoty, priemernej hodnoty a štandardnej odchýlky. Pre lepšie grafické znázornenie si transponujeme popis pomocou metódy **transpose** [11]. Po spustení príkazu dostaneme nasledujúci výpis pre trénovacie dáta:

	count	mean	std	min	25%	50%	75%	max
type	3318.0	0.2510548523206751	0.4336853687878535	0.0	0.0	0.0	1.0	1.0
fixed acidity	3318.0	7.200649668312325	1.307856289689693	3.8	6.4	7.0	7.7	15.9
volatile acidity	3318.0	0.3436448807619828	0.16740229641933257	0.08	0.23	0.29	0.41	1.58
citric acid	3318.0	0.3149163375782691	0.14609589162402678	0.0	0.24	0.31	0.39	1.66
residual sugar	3318.0	5.121371754730083	4.591229880967519	0.6	1.8	2.8	7.6	65.8
chlorides	3318.0	0.05630615512783249	0.035975081953832665	0.009	0.037	0.047	0.066	0.61
free sulfur dioxide	3318.0	30.25015069318867	18.20041655498559	1.0	16.0	28.0	41.0	289.0
total sulfur dioxide	3318.0	114.16742013261	56.71644226384385	6.0	75.0	116.0	154.0	440.0
density	3318.0	0.9945327034358048	0.003023173813494118	0.98711	0.9921425	0.994635	0.9967975	1.03898
pH	3318.0	3.226331379550249	0.15922283965090384	2.72	3.12	3.22	3.33	4.01
sulphates	3318.0	0.5290866112728486	0.14662963829553466	0.22	0.43	0.51	0.6	2.0
alcohol	3318.0	10.562140847896021	1.1827081072713241	8.0	9.5	10.4	11.4	14.9

Tabuľka 3: Neškálovaný výstup metódy **describe().transpose()** na trénovacích dátach

	count	mean	std	min	25%	50%	75%	max
type	1106.0	0.2640144665461121	0.44100643408418866	0.0	0.0	0.0	1.0	1.0
fixed acidity	1106.0	7.2302893309222425	1.2972125728647117	3.9	6.4	7.0	7.8	13.4
volatile acidity	1106.0	0.3423598553345389	0.16636366635167069	0.1	0.23	0.3	0.4	1.24
citric acid	1106.0	0.3223056057866185	0.15187938982798016	0.0	0.24	0.31	0.4	1.0
residual sugar	1106.0	5.079169513738172	4.514410888526974	0.7	1.8	2.7	7.7	22.0
chlorides	1106.0	0.05624486683015209	0.03414465787609784	0.0140	0.039	0.047	0.066	0.464
free sulfur dioxide	1106.0	29.96745027124774	17.516891090972734	2.0	17.0	28.0	40.0	138.5
total sulfur dioxide	1106.0	114.24141048824593	56.801217590987235	8.0	76.0	117.0	152.0	313.0
density	1106.0	0.9945598734177216	0.0029059188206326794	0.98713	0.9923	0.99472	0.9968	1.00242
pH	1106.0	3.220297731167069	0.1641068028002486	2.83	3.11	3.21	3.32	3.85
sulphates	1106.0	0.5396202531645569	0.15023071399198695	0.26	0.44	0.52	0.61	1.59
alcohol	1106.0	10.529355033160941	1.2018754181703872	8.4	9.5	10.3	11.4	14.2

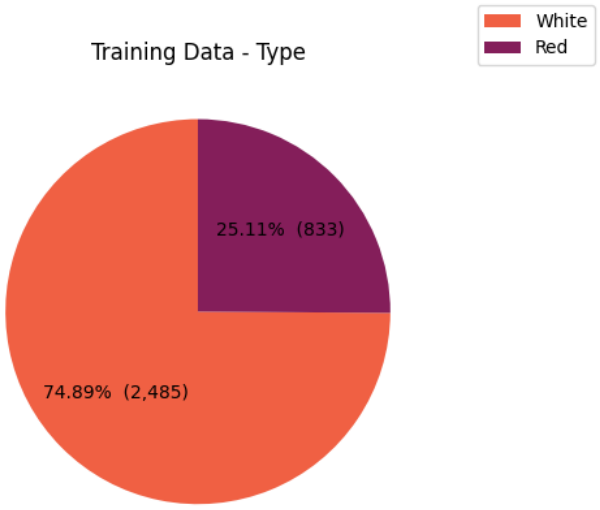
Tabuľka 4: Neškálovaný výstup metódy **describe().transpose()** na validačných dátach

Potom to spustíme aj pre trénovacie dáta:

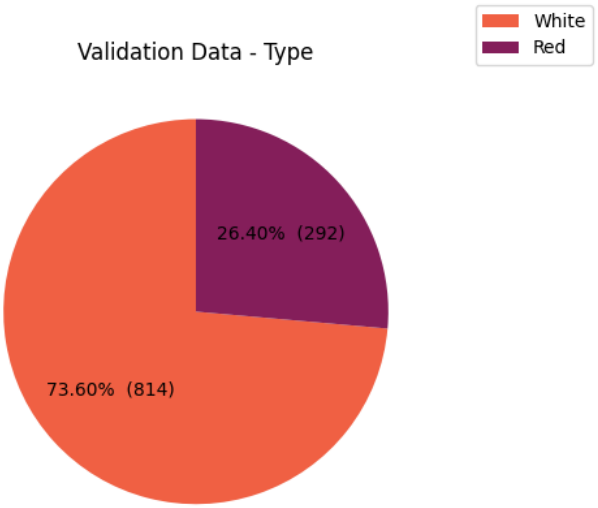
	count	mean	std	min	25%	50%	75%	max
type	1238.0	0.2495961227786753	0.4329541676511395	0.0	0.0	0.0	0.0	1.0
fixed acidity	1238.0	7.2622828576072775	1.3424577916846911	4.2	6.4	7.0	7.7	15.6
volatile acidity	1238.0	0.34203491811062625	0.16684691518826036	0.08	0.23	0.3	0.41	1.33
citric acid	1238.0	0.32930226497766435	0.14419202487865299	0.0	0.26	0.32	0.41	0.81
residual sugar	1238.0	5.26175282714055	4.596489688736631	0.6	1.8	2.9	7.7	26.05
chlorides	1238.0	0.05706946688206784	0.039785500316662656	0.012	0.038	0.0475	0.066	0.611
free sulfur dioxide	1238.0	29.70193861066236	17.108066697248095	3.0	16.0	28.0	40.875	105.0
total sulfur dioxide	1238.0	115.46768982229402	56.613913235720375	6.0	78.0	117.5	155.0	344.0
density	1238.0	0.9946835864297252	0.0028985659873016745	0.98742	0.9924	0.9948	0.9969	1.00315
pH	1238.0	3.215395799676898	0.1599539041778244	2.85	3.1	3.21	3.32	3.9
sulphates	1238.0	0.5379481357574883	0.15756224993035925	0.25	0.44	0.51	0.6	1.98
alcohol	1238.0	10.481965535794023	1.1932810323841043	8.4	9.5	10.3	11.3	14.0

Tabuľka 5: Neškálovaný výstup metódy **describe().transpose()** na testovacích dátach

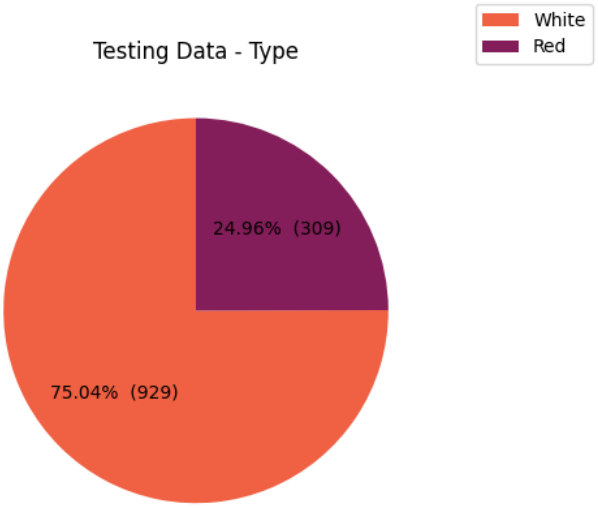
Pre zaujímavosť si môžeme vykresliť aj graf, ktorý znázorní pomer bieleho víne ku červenému.



Obr. 1: Pomer vín v trénovacích dátach



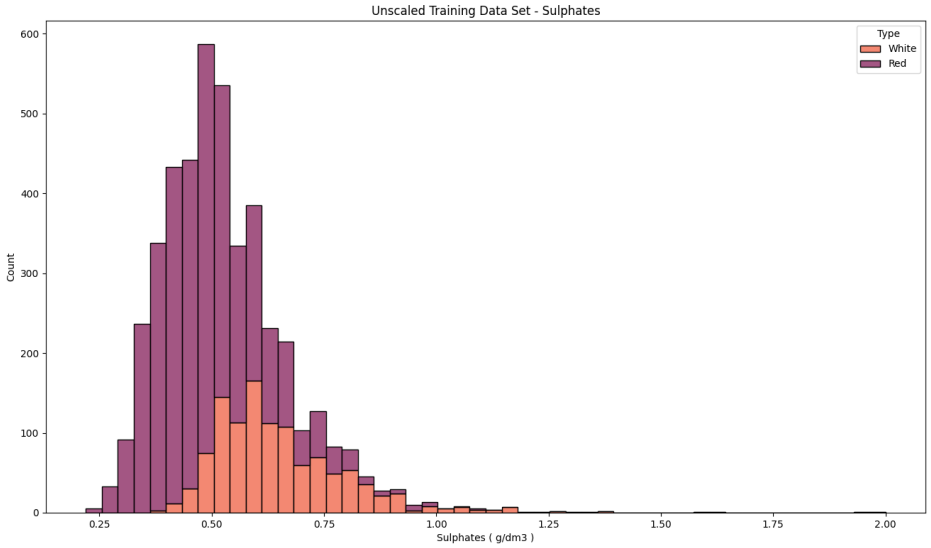
Obr. 2: Pomer vín vo validačných dátach



Obr. 3: Pomer vín v testovacích dátach

Nasledovne, si môžeme vykresliť aj histogram pre jednotlivé stĺpce, z ktorých na ukážku som si vybral stĺpec ‘sulphates‘ a pre zaujímavosť si zafarbím na základe typu vína.

Na grafe vidíme, že stĺpec obsahuje outlier hodnoty, ktoré sú veľmi ďaleko do prava (okolo hodnoty 2.00). Tieto hodnoty v niektorých prípadoch môžu znižovať úspešnosť našej



Obr. 4: Stĺpec Sulphates s názorneným typom vína

siete, ale napriek tomu ich nevyhodíme. Teraz nasleduje škálovanie dát, na čo použijeme **StandardScaler** [12]. Po škálovaní si spustíme tie isté príkazy a pozrieme sa na dáta.

	count	mean	std	min	25%	50%	75%	max
type	3318.0	9.529581597746974e-17	1.000150727259881	-0.5789743237878807	-0.5789743237878807	-0.5789743237878807	1.7271923104596445	1.7271923104596445
fixed acidity	3318.0	-2.3984564920172156e-16	1.0001507272598809	-2.6005626655858474	-0.6122770172500739	-0.15344186763412646	0.38186580691781263	6.652612851669097
volatile acidity	3318.0	-8.458842092382145e-17	1.000150727259881	-1.5751553286457178	-0.6789752146454616	-0.32050316904535936	0.3964409221548455	7.3866458113568445
citric acid	3318.0	-4.282958021459314e-17	1.0001507272598809	-2.1558703708484392	-0.5128660955461808	-0.03365651524968863	0.514011576517731	9.208242533325516
residual sugar	3318.0	2.0344050601931741e-16	1.0001507272598809	-0.984932875491879	-0.7235256047108872	-0.5056862123933941	0.5399428707305731	13.218195503608674
chlorides	3318.0	-1.3705465668669804e-16	1.0001507272598809	-1.315168246612703	-0.536734429583047	-0.2587223520724555	0.2695005951976684	15.393357611773846
free sulfur dioxide	3318.0	-5.3536975268241427e-17	1.0001507272598809	-1.6073565898882012	-0.7830754057907234	-0.12365045851769264	0.5907265676947574	14.218842144670727
total sulfur dioxide	3318.0	2.5697748128755883e-17	1.0001507272598809	-1.9074490499278192	-0.6906872534121516	0.0323161329232448446	0.7024168324536124	5.745806307866379
density	3318.0	-3.2362030810146576e-14	1.0001507272598809	-2.45563857639209	-0.7907463652763862	0.03384257386040226	0.7492582532519081	14.70441288720892
pH	3318.0	-1.6548279055413425e-15	1.0001507272598809	-3.1804965832915637	-0.6679155253159325	-0.03977026082202398	0.6511895301212739	4.9225773286798455
sulphates	3318.0	-4.036687935225403e-16	1.0001507272598809	-2.1082586211374825	-0.675862993854733	-0.13018846917559035	0.48369537108844485	10.03299955297344
alcohol	3318.0	3.99385835501081e-16	1.0001507272598809	-2.1666605788959776	-0.8981936751296238	-0.1371135328698113	0.7085310696410911	3.66828717842925

Tabuľka 6: Škálovaný výstup metódy **describe().transpose()** na trénovacích dátach

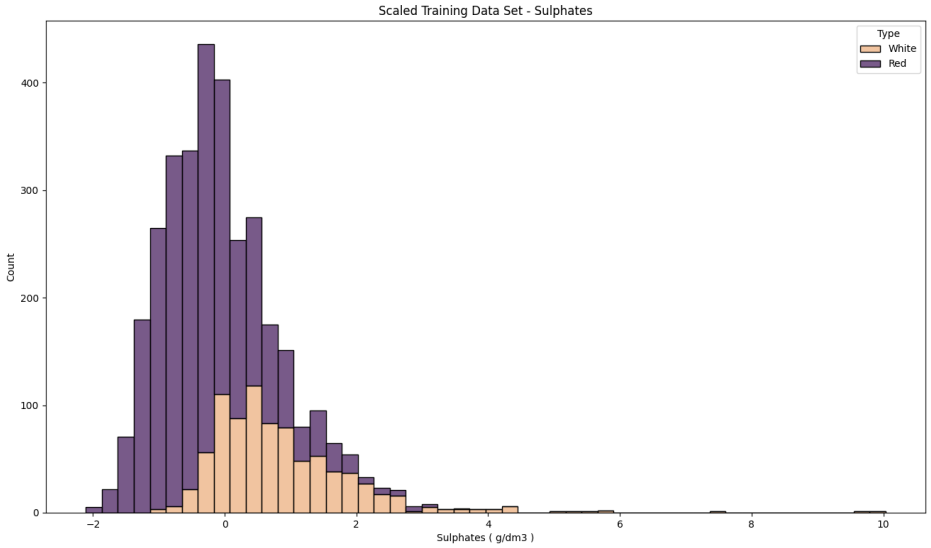
	count	mean	std	min	25%	50%	75%	max
type	1106.0	0.029887029919422502	1.0170343237734363	-0.5789743237878807	-0.5789743237878807	-0.5789743237878807	1.7271923104596445	1.7271923104596445
fixed acidity	1106.0	0.022666198380312787	0.9920112082567807	-2.5240901406498564	-0.6122770172500739	-0.15344186763412646	0.4583383318538037	4.740799728269316
volatile acidity	1106.0	-0.007677428227066109	0.993945396510272	-1.455664646779017	-0.6789752146454616	-0.2607578281120089	0.33669558122149534	5.35530421962293
citric acid	1106.0	0.05058583024030437	1.039743695073624	-2.1558703708484392	-0.5128660955461808	-0.03365651524968863	0.5824700879886585	4.689980776244305
residual sugar	1106.0	-0.009193310532114434	0.9834165246281903	-0.9631489362601297	-0.7235256047108872	-0.5274701516251433	0.5617268099623225	3.6768301201024745
chlorides	1106.0	-0.0017038886965216372	0.9492627272022341	-1.17616220785741	-0.48113201408092865	-0.2587223520724555	0.2695005951976684	11.334381280119207
free sulfur dioxide	1106.0	-0.015534975902705063	0.9625895820043432	-1.5524045109425928	-0.7281233268513042	-0.12365045851769264	0.5357744887553382	5.948554264288132
total sulfur dioxide	1106.0	0.0013047628702667084	1.0016456747867675	-1.8721805920577996	-0.673053024477142	0.0499503618582545 15	0.667148374583593	3.5062592331201503
density	1106.0	0.00898859240323894	0.9613594854656556	-2.449022015195436	-0.7386409458525132	0.0619629589463299 43	0.750085323401499	2.609339019669841
pH	1106.0	-0.03790007659515654	1.0308291105021177	-2.4895367923482654	-0.7307300517653247	-0.10258478727141622	0.5883750036718817	3.917544905489595
sulphates	1106.0	0.07184925040497821	1.0247134181223212	-1.835421358797911	-0.6076536782698402	-0.06197915359069747	0.5519046866733377	7.236417613992836
alcohol	1106.0	-0.027725147269643594	1.0163594602663217	-1.8284027378916163	-0.8981936751296238	-0.22167799312090125	0.7085310696410911	3.0763359566716173

Tabuľka 7: Škálovaný výstup metódy **describe().transpose()** na validačných dátach

	count	mean	std	min	25%	50%	75%	max
type	1238.0	-0.0033640733981510768	0.998464455595467	-0.5789743237878807	-0.5789743237878807	-0.5789743237878807	-0.5789743237878807	1.7271923104596445
fixed acidity	1238.0	0.047132456052429834	1.026611369501233	-2.294672565841882	-0.6122770172500739	-0.15344186763412646	0.38186580691781263	6.423195276861123
volatile acidity	1238.0	-0.009618776749525663	0.9968325831600416	-1.5751553286457178	-0.6789752146454616	-0.2607578281120089	0.3964409221548455	5.893012288023084
citric acid	1238.0	0.09848391758914268	0.9871171389171519	-2.1558703708484392	-0.37594907260432575	0.034801996221238876	0.6509285994595857	3.389269058296684
residual sugar	1238.0	0.03058052750677403	1.0012965205880109	-0.984932875491879	-0.7235256047108872	-0.48390227316164475	0.5617268099623225	4.559079658988322
chlorides	1238.0	0.021220988658322386	1.106084959783368	-1.2317646233595256	-0.5089332218319879	-0.24482174819692587	0.2695005951976684	15.421158819524901
free sulfur dioxide	1238.0	-0.030125393634529218	0.9401238316480267	-1.4974524320031735	-0.7830754057907234	-0.12365045851769264	0.58385755782733	4.107659619817588
total sulfur dioxide	1238.0	0.02292925338514196	0.9983427069054673	-1.9074490499278192	-0.6377845666071227	0.05876747632575935	0.7200510613886221	4.05292033010545
density	1238.0	0.04991632814071862	0.9589269618804749	-2.3530818778435076	-0.7055581398690948	0.08842920373305732	0.7831681293849174	2.850843503348788
pH	1238.0	-0.06869132711940219	1.0047428744661273	-2.363907739449484	-0.7935445782147141	-0.10258478727141622	0.5883750036718817	4.231617537736548
sulphates	1238.0	0.06044385201360422	1.074721322976563	-1.9036306743828038	-0.6076536782698402	-0.13018846917559035	0.4836953710884444 85	9.896580921803654
alcohol	1238.0	-0.067799819933682 17	1.0090916643142551	-1.8284027378916163	-0.8981936751296238	-0.22167799312090125	0.6239666093900013	2.9072070361694373

Tabuľka 8: Škálovaný výstup metódy **describe().transpose()** na testovacích dátach

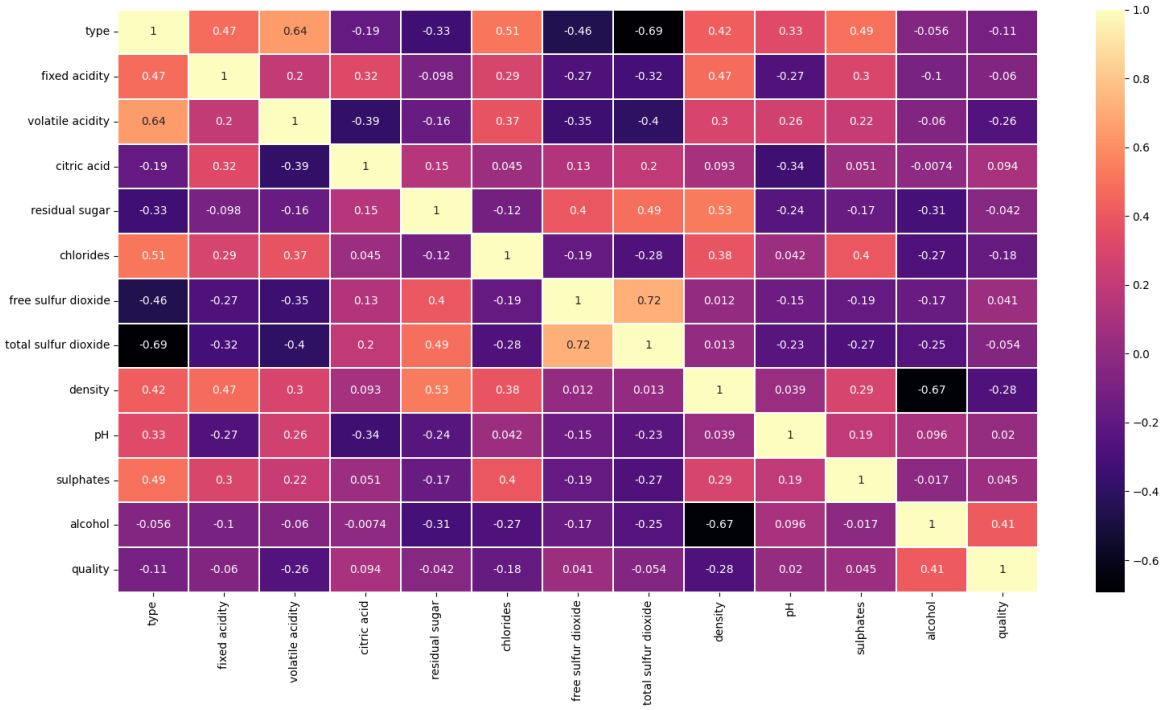
V dátach vidíme, že na trénovacích dátach sme použili fit + transform (alebo fit_transform) a na ostatných len transform. Nemôžeme používať fit na validačných a testovacích dátach, aby náhodou nenastal Data Leakage [13], aby nám program nevedel dopredu aké sú presné dáta.



Obr. 5: Stĺpec Sulphates so znázorneným typom vína po škálovaní

Na obrázku vidíme, že už po škálovaní nám zmenilo počet hodnôt (y), kvôli rozdeleniu na trénovacie a validačné dáta, a samotné hodnoty (x) sa presunuli tak, aby vrchol štandardnej odchýlky bola v nule.

Ďalej si môžeme naštudovať aké sú korelácie medzi jednotlivými stĺpcami. Na obrázku vidíte Heatmap čo sa často používa na takéto účely. Je to farebne aj číselne znázornené, že v akej silnej korelácii sú jednotlivé stĺpce so sebou.



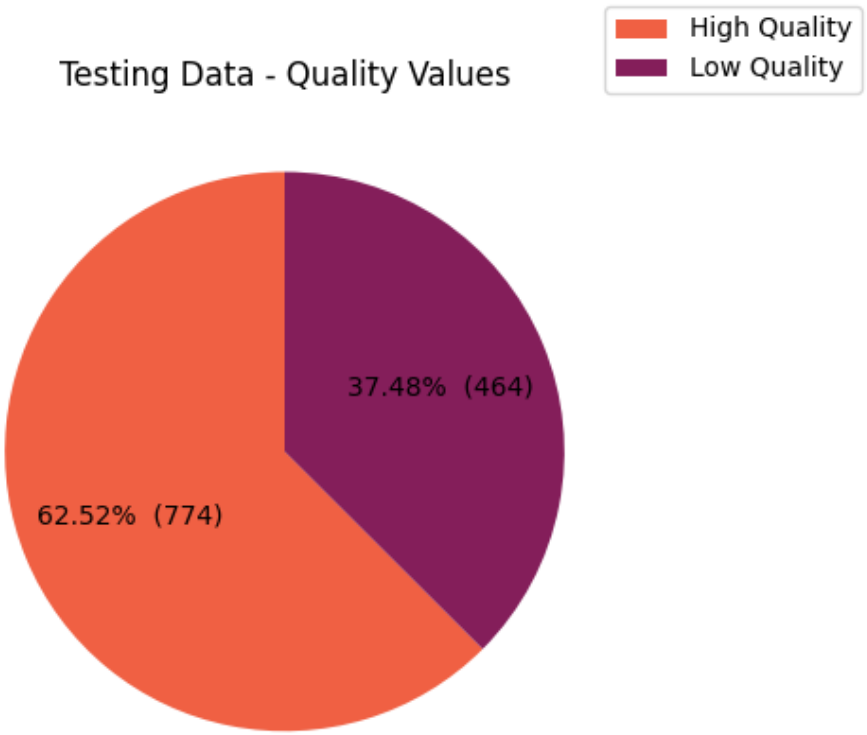
Obr. 6: Heatmap pre naše dáta

Máme aj menej farebnú verziu (teraz pre stĺpec quality). Tieto hodnoty ešte budeme skúmať neskoršie.

```
density              -0.276095
volatile acidity     -0.258214
chlorides            -0.180525
type                 -0.109787
fixed acidity        -0.060167
total sulfur dioxide -0.053554
residual sugar       -0.041768
pH                    0.020164
free sulfur dioxide   0.040971
sulphates             0.045478
citric acid           0.094210
alcohol               0.414508
quality               1.000000
```

2.3 Náhodný Klasifikátor (Random Classifier)

Túto úlohu začneme s tým, že sa pozrieme na to, že v akom pomere sú v testovacích dátach hodnoty kvality. Toto vidíte na tomto grafe:



Obr. 7: Pomer hodnôt v stĺpci quality v testovacích dátach

Na výpočet používame privátnu metódu **NeuralNetworkProject.___calculate_random_classifier_accuracy** (Súbor: neural_network_project.py), ktorá vyzerá nasledovne:

```
@staticmethod
def __calculate_random_classifier_accuracy(
        data: pd.DataFrame,
        column: str,
        title: str
) -> None:
    """Calculate Accuracy for Random Classifier.

    Args:
        data (pandas.DataFrame) : Data to work with
        column (str) : Column to process
        title (str) : Title for information

    """

    # Get the number of rows in the DataFrame
    total_rows = len(data)

    # Get the Unique Values for the given column
    unique_values = data[column].value_counts()

    # Create empty list where the components of the equation will be stored
    unique_values_percentage = list()

    # For each unique value
    for value in unique_values:
        # Calculate the percentage of the given values and square it
        unique_values_percentage.append(round(value / total_rows, 4) ** 2)

    # Display the Final Accuracy with a custom message
    print(f"{colored(f'{title}:', 'green')} "
          f"{sum(unique_values_percentage)}")
```

Na základe vzorca hodnota bude nasledovná

```
acc= 0.6252^2 + 0.3748^2 = 0.39087504 + 0.14047504 = 0.53135008 => 53.13%
```

2.4 Logistická Regresia (Logistic Regression)

Po škálovaní a rozdelení (na vstupné X a výstupné y dáta) si spustíme na trénovacích a testovacích dátach Logistický Regresný model, ktorý nám produkuje nasledujúce výsledky. Ak si pozrieme na hodnotu accuracy, čo je v našom prípade 0.74 to znamená, že Logistický Regresný Model bol schopný na 74% správne klasifikovať vzorky v testovacích dátach. Táto hodnota je už oveľa lepšie ako 53.13% v prípade Náhodného Klasifikátora.

Classification Report:

	precision	recall	f1-score	support
0	0.67	0.61	0.64	464
1	0.78	0.82	0.80	774
accuracy			0.74	1238
macro avg	0.72	0.71	0.72	1238
weighted avg	0.74	0.74	0.74	1238

Na Logistickú Regresiu používame privátnu metódu **NeuralNetworkProject.___run_logistic_regression** (Súbor: neural_network_project.py), ktorá vyzerá nasledovne:

```
# noinspection PyPep8Naming
@staticmethod
def __run_logistic_regression(
        X_train: pd.DataFrame,
        y_train: pd.Series,
        X_test: pd.DataFrame,
        y_test: pd.Series,
        randoms_tate: int,
        max_iter: int
) -> None:
    """Run Logistic Regression Process ( Create, Train, Predict, Evaluate ).

    Args:
        X_train (pandas.DataFrame) : Training Feature Set
        y_train (pandas.Series) : Training Label Set
        X_test (pandas.DataFrame) : Testing Feature Set
        y_test (pandas.Series) : Testing Label Set
        randoms_tate (int) : Random State for Model
        max_iter (int) : Max Number of Iteration for Model

    """

    # Inform the User
    print(colored('Logistic Regression\n', 'green'))

    # Create Logistic Regression Model
    logistic_regression = LogisticRegression(
        random_state=randoms_tate,
        max_iter=max_iter
    )
```

```
# Train the Logistic Regression Model
logistic_regression.fit(X=X_train, y=y_train)

# Predict using the Logistic Regression Model
logistic_regression_prediction = logistic_regression.predict(X=X_test)

# Display Classification Report
print(colored('Classification Report:', 'green'))
print(
    classification_report(
        y_true=y_test,
        y_pred=logistic_regression_prediction
    )
)
```

2.5 Neurónová Sieť

2.5.1 Grid Search

Grid Search je metóda, ktorá sa používa na vyhľadanie najoptimálnejších nastavení pre neurónové siete. V tomto zadaní, sme nevyužili už vopred vytvorenú metódu **GridSearchCV** [14], ale namiesto toho sme si vytvorili vlastnú metódu. V rámci našej metódy, sme používali nasledujúce kritéria:

```
learning_rates = [0.001, 0.0001, 0.00001]
activation_functions = ['sigmoid', 'relu']
layers = [2, 3, 4, 5]
neurons = [16, 32, 64, 128]
batch_sizes = [512, 1024, len(X_train)]
patience_list = [25, 40, 55]
```

Kritérium **learning_rates** reprezentuje rýchlosť učenia, ktorý využijeme v Adam optimizeri. **Activation_function** reprezentuje aktivačnú funkciu, ktorú bude obsahovať každý neurón v sieti okrem výstupnej vrstvy. V tomto zadaní, sme použili chybovú funkciu **binary_crossentropy** [15], ktorá sa používa na binárne klasifikačné úlohy, a tým pádom v neuróne vo výstupnej vrstve si musíme nastaviť aktivačnú funkciu **sigmoid** [16], aby nám generoval správne výsledky. **Layers** predstavuje počet skrytých vrstiev v sieti, **neurons** reprezentuje počet neurónov v jednej vrstve (naša implementácia nepodporuje rôzny počet neurónov v jednotlivých vrstvách). **Batch_size** reprezentuje veľkosť jedného batchu, s ktorým pracuje naša sieť počas trénovania. **Patience_list** predstavuje možné hodnoty, nad ktorými uvažujeme pri vytváraní **EarlyStopping** callbacku [17].

```
Grid Search Completed in: 15:34:25.535020
```

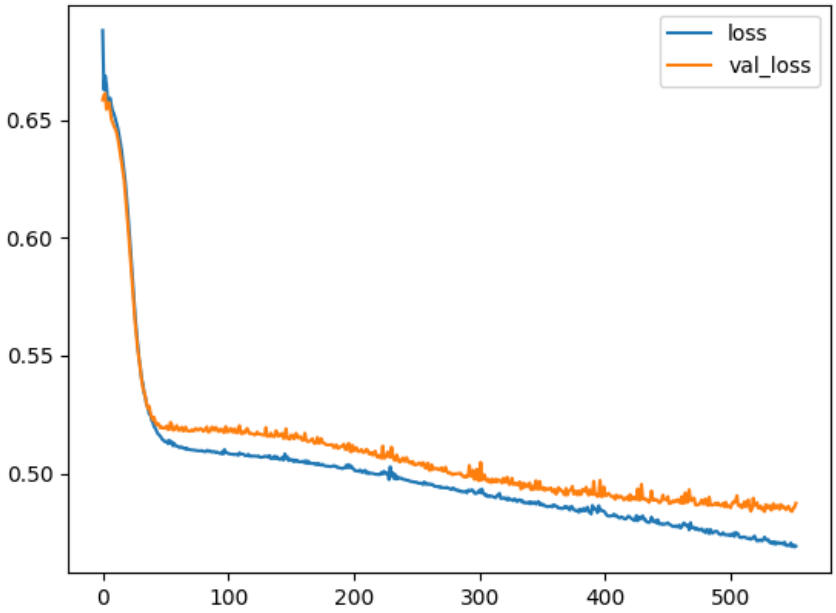
Potom pomocou tejto metódy sme našli konfiguráciu, ktorým sme dosiahli 79 percentnú úspešnosť.

Learning Rate	Activation Function	Layers	Neurons	Batch Size	Patience
0.001	Relu	3	64	1024	25

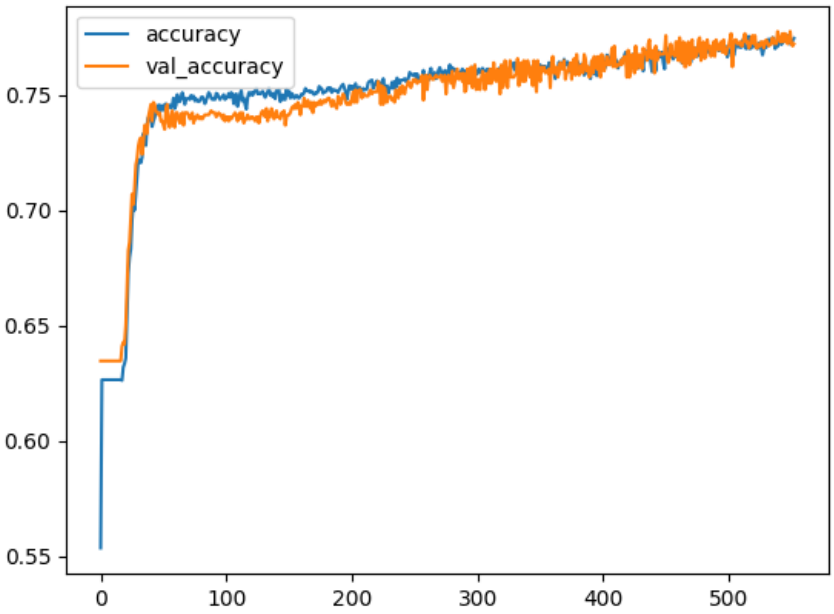
Tabuľka 9: Výstup metódy **NeuralNetwork.grid_search**

2.5.2 Finálna štruktúra

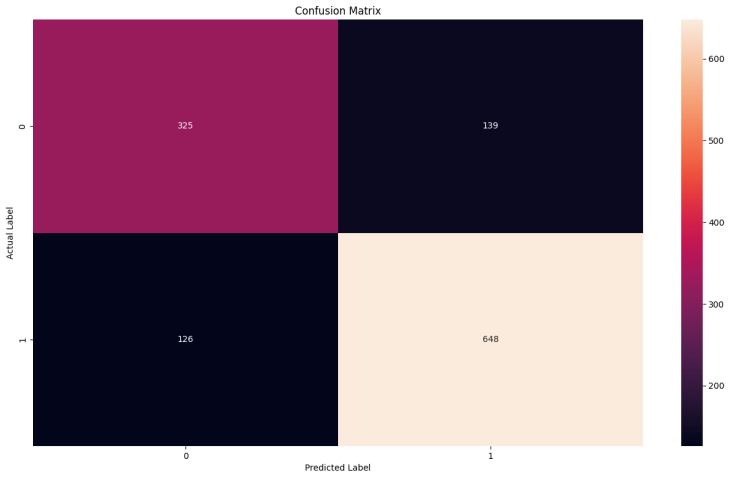
Pre finálnu štruktúru sme si vybrali najlepšiu výstupnú konfiguráciu z metódy **grid_search**. Nižšie vidíte dosiahnuté výsledky. Na trénovanie sme používali 10000 epoch a vidíme, že **Early Stopping** bol zavolaný pri 550. epoche. Grafy sú minimálne zašumené a to kvôli rýchlosti učenia. Ale napriek tomu táto sieť dosiahla najlepšie výsledky. 265 vzoriek z 1238 klasifikovala nesprávne. Táto sieť vyhovuje na klasifikáciu vína. Ostatné výsledky metódy Grid Search nájdete v súbore **grid_search_output.txt**.



Obr. 8: Vývoj chybovej hodnoty počas trénovania



Obr. 9: Vývoj úspešnosti počas trénovania



Obr. 10: Konfúzna matica

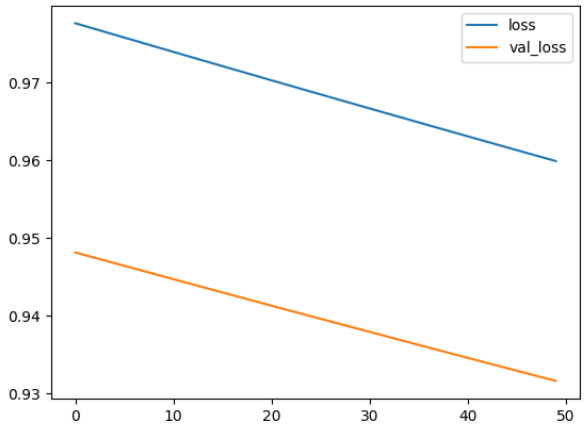
Classification Report:

	precision	recall	f1-score	support
0	0.72	0.70	0.71	464
1	0.82	0.84	0.83	774
accuracy			0.79	1238
macro avg	0.77	0.77	0.77	1238
weighted avg	0.78	0.79	0.79	1238

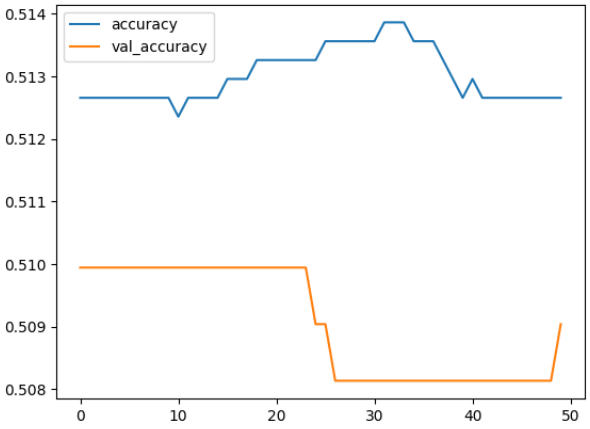
2.5.3 Modifikácia štruktúry

Po nájdení najoptimálnejšej štruktúry sme ju trochu modifikovali. Na nasledujúcich obrázkoch vidíte príklad podtrénovania. Vidíme, že naša sieť s týmito konfiguráciami dosiahla 52 percentnú úspešnosť, čo je slabšia ako Náhodný Klasifikátor. Sieť v tomto krátkom čase nevedel poriadne natrénovať s takýmto nízkym počtom neurónov a s vypnutým Early Stoppingom. Podarilo sa mu trochu znížiť chybovú hodnotu na konci trénovania, ale napriek tomu, nevyhovuje na spoľahlivú kvalifikáciu vína. Tieto hodnoty sme dosiahli so štruktúrou:

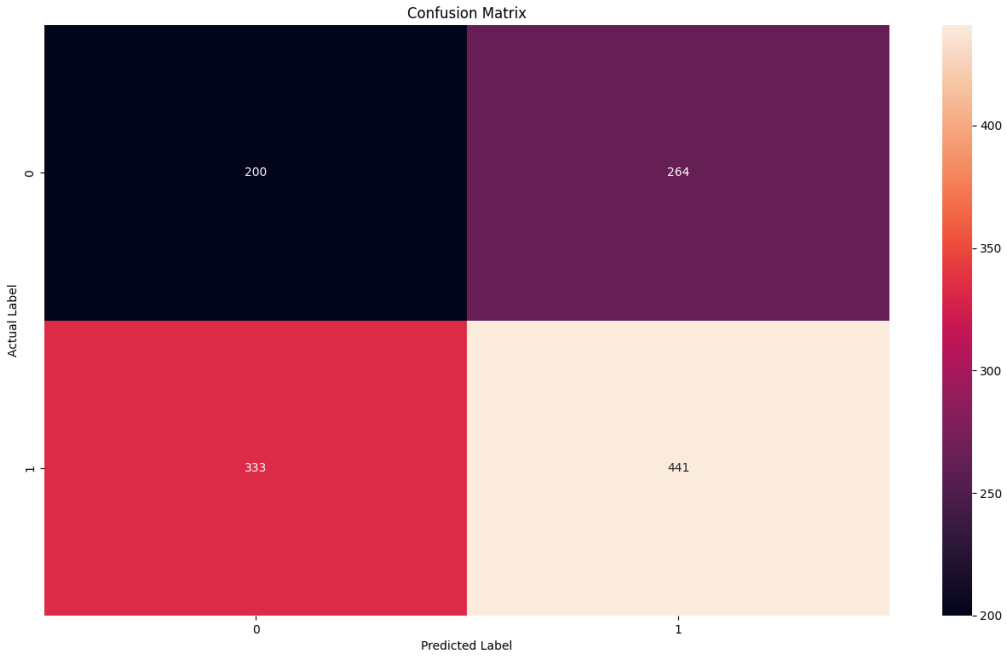
Learning Rate	Activation Function	Layers	Neurons	Batch Size	Patience	Max Epochs
0.0001	Relu	2	2	len(X_trian)	Vypnutý ES	50



Obr. 11: Vývoj chybovej hodnoty počas trénovania



Obr. 12: Vývoj úspešnosti počas trénovania



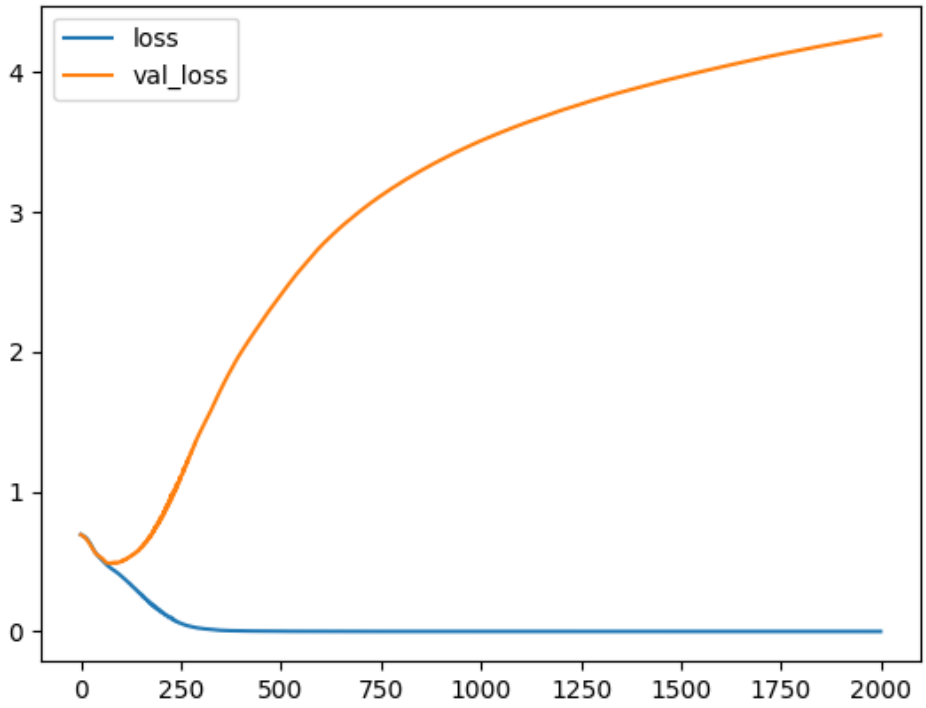
Obr. 13: Konfúzna matica podtrénovanej siete

Classification Report:

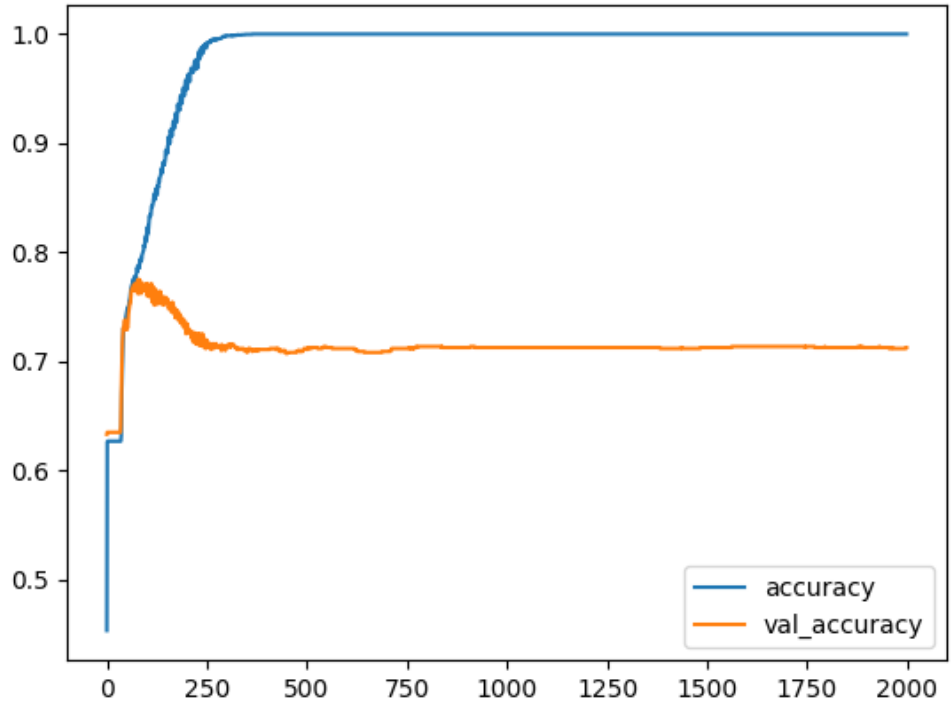
	precision	recall	f1-score	support
0	0.38	0.43	0.40	464
1	0.63	0.57	0.60	774
accuracy			0.52	1238
macro avg	0.50	0.50	0.50	1238
weighted avg	0.53	0.52	0.52	1238

Na nasledujúcich obrázkoch vidíte príklad pretrénovania. Vidíme, že naša sieť v tomto časovom intervale už bol schopný natrénovať, ale samotné trénovanie už by malo skončiť skorej. V tomto prípade sme tiež vypli Early Stopping. Na obrázku vidíme, že na validačných dátach máme nesmierne veľkú chybovú hodnotu, čo je typickým príznakom pretrénovania. Pravdepodobne trénovanie by malo zastaviť pri 100-150. epoche, aby sme nemali takúto príliš veľkú chybovú hodnotu. V porovnaní na validačných dátach naša sieť dosiahla príliš nízku úspešnosť v porovnaní s úspešnosťou na trénovacích dátach. Napriek tomu, že sieť dosiahla 78 percentnú úspešnosť na trénovacích dátach (čo môže byť aj čistá náhoda), ktorá je síce trochu väčšia ako úspešnosť Logistickej Regresie, túto sieť by som neodporúčal na klasifikáciu, z toho dôvodu, že z pohľadu strojového učenia nie je vhodne nakonfigurovaná a preto na nevidených dátach môže produkovať nepresné výsledky. Tieto hodnoty sme dosiahli so štruktúrou:

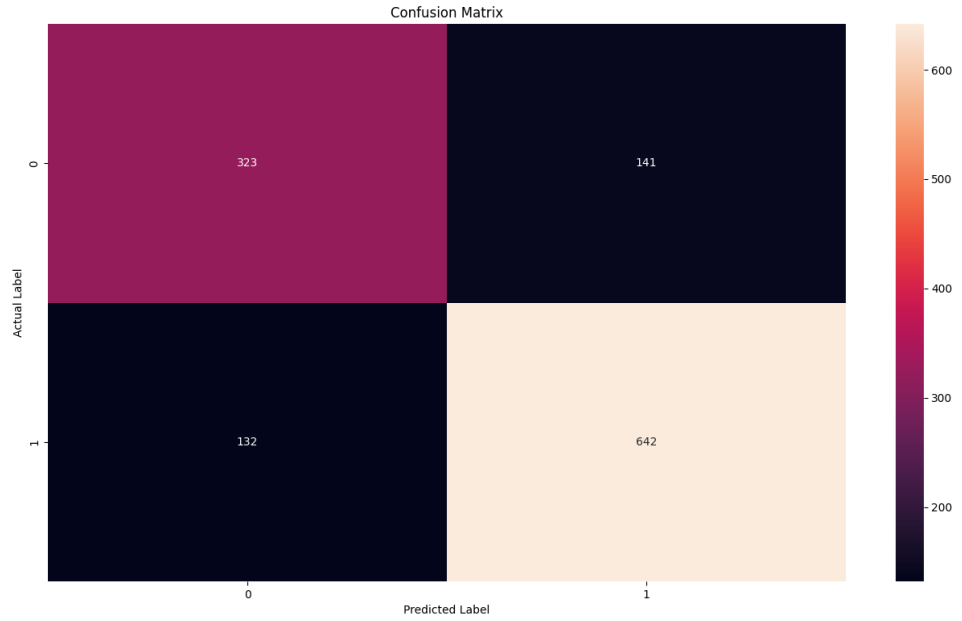
Learning Rate	Activation Function	Layers	Neurons	Batch Size	Patience	Max Epochs
0.0001	Relu	10	256	len(X_train)	Vypnutý ES	2000



Obr. 14: Vývoj chybovej hodnoty počas trénovania



Obr. 15: Vývoj úspešnosti počas trénovania



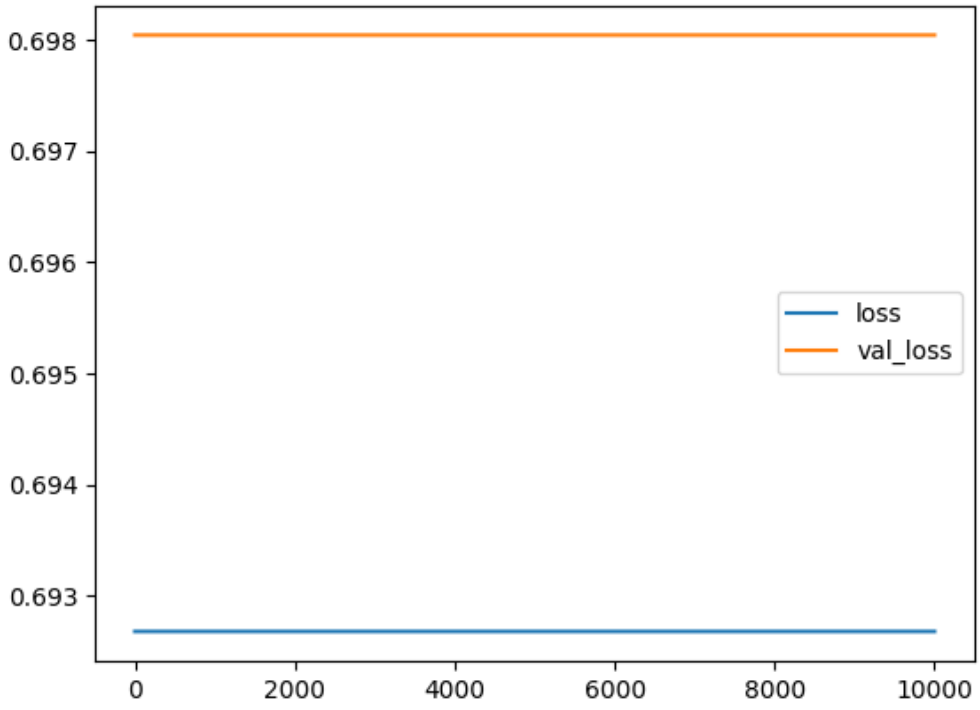
Obr. 16: Konfúzna matica pretrénovanej siete

Classification Report:

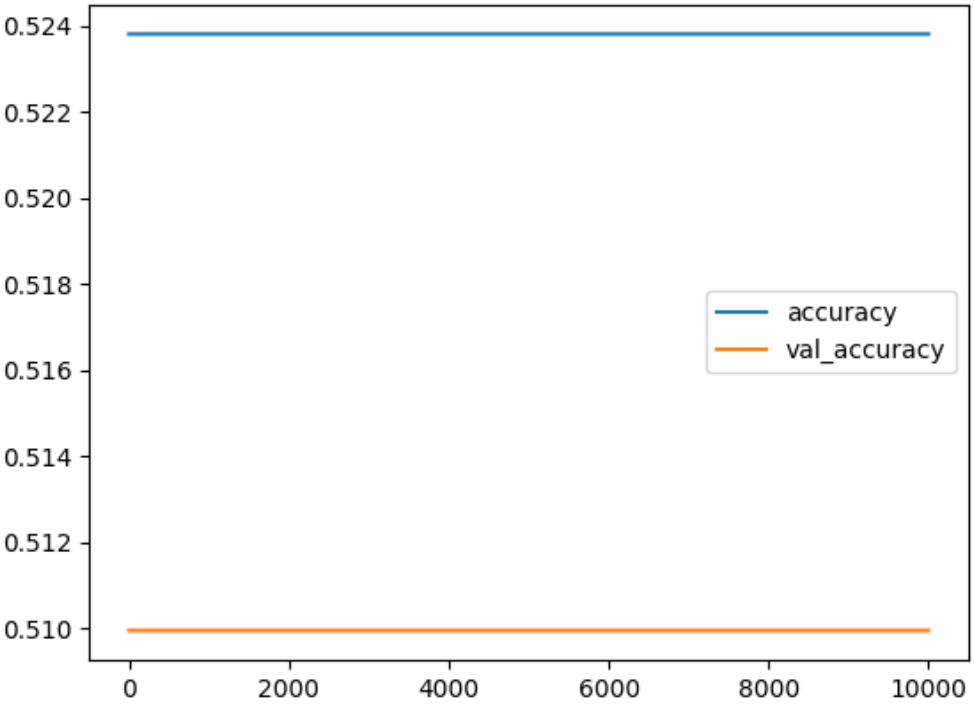
	precision	recall	f1-score	support
0	0.71	0.70	0.70	464
1	0.82	0.83	0.82	774
accuracy			0.78	1238
macro avg	0.76	0.76	0.76	1238
weighted avg	0.78	0.78	0.78	1238

Potom sme vyskúšali ako bude vyzerať výstup, ak spomalíme proces učenia. Výsledky vidíte na nižšie uvedených obrázkoch. Vidíme, že vývoj chybovej hodnoty na trénovacích dátach vyvíja strašne pomaly, a tým pádom, chybové hodnoty na validačných dátach sú veľmi vysoké. V tomto prípade sme vypli **Early Stopping**, aby sme mohli dosiahnuť čo najpresnejšie výsledky. Úspešnosť na validačných dátach v porovnaní s úspešnosťou na trénovacích dátach je celkom nízka. Sieť nie je vhodne nakonfigurovaná. Tieto hodnoty sme dosiahli so štruktúrou:

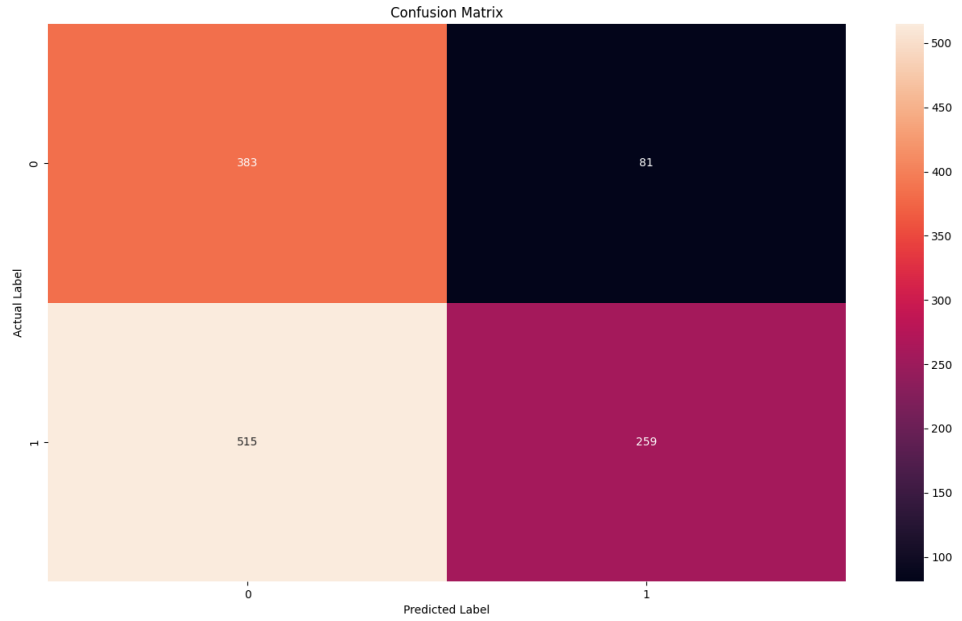
Learning Rate	Activation Function	Layers	Neurons	Batch Size	Patience	Max Epochs
0.00000000000000000000000000001	Relu	3	64	1024	Vypnutý ES	10000



Obr. 17: Vývoj chybovej hodnoty počas trénovania



Obr. 18: Vývoj úspešnosti počas trénovania



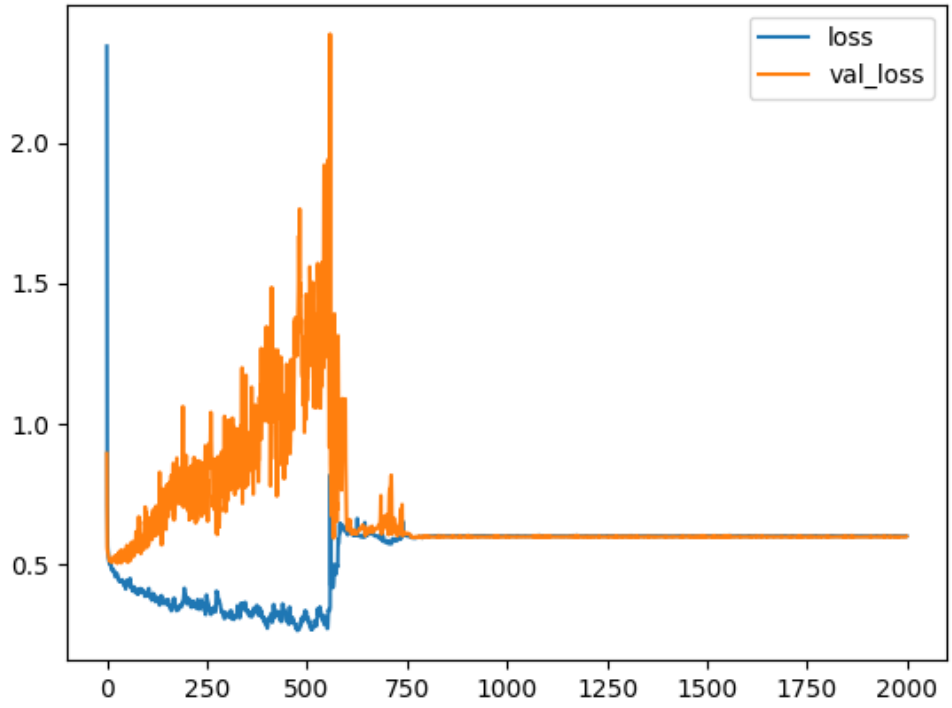
Obr. 19: Konfúzna matica (Low Learning Rate)

Classification Report:

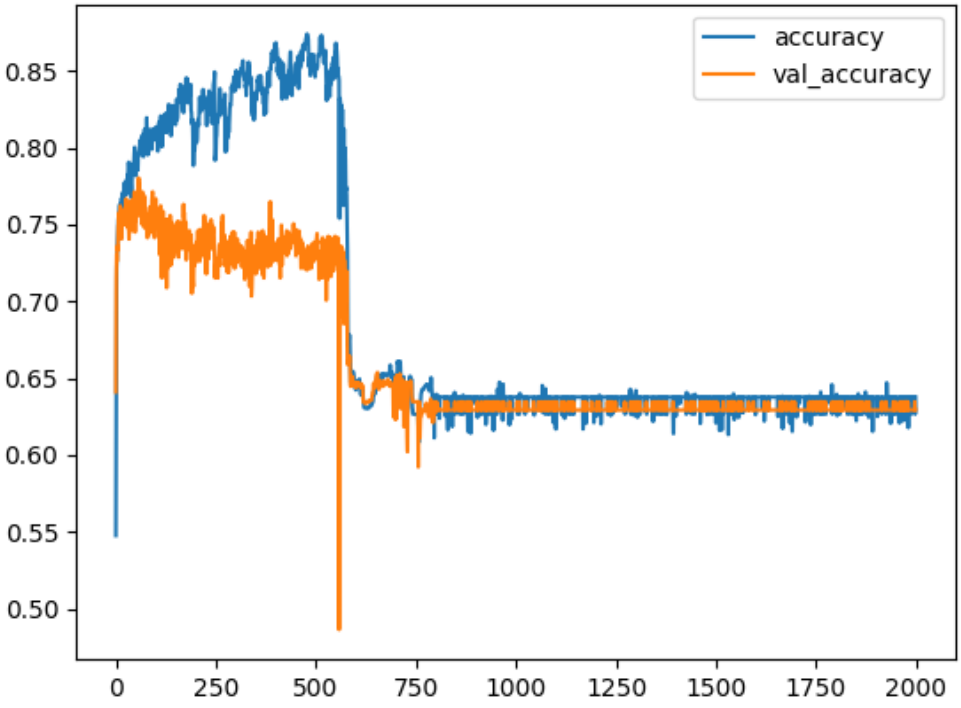
	precision	recall	f1-score	support
0	0.43	0.83	0.56	464
1	0.76	0.33	0.46	774
accuracy			0.52	1238
macro avg	0.59	0.58	0.51	1238
weighted avg	0.64	0.52	0.50	1238

Potom sme vyskúšali ako bude vyzerať výstup, ak zrýchlime proces učenia. Výsledky vidíte na nižšie uvedených obrázkoch. Táto rýchlosť nedovoľuje sieti, aby vykonal menšie modifikácie, a kvôli tomu vidíme, že po vykonaní radikálnych zmien chybová hodnota rýchlo stúpa na validačných dátach. Sieť na klasifikáciu vína by som neodporúčal, kvôli nesprávnej konfigurácie. Tieto hodnoty sme dosiahli so štruktúrou:

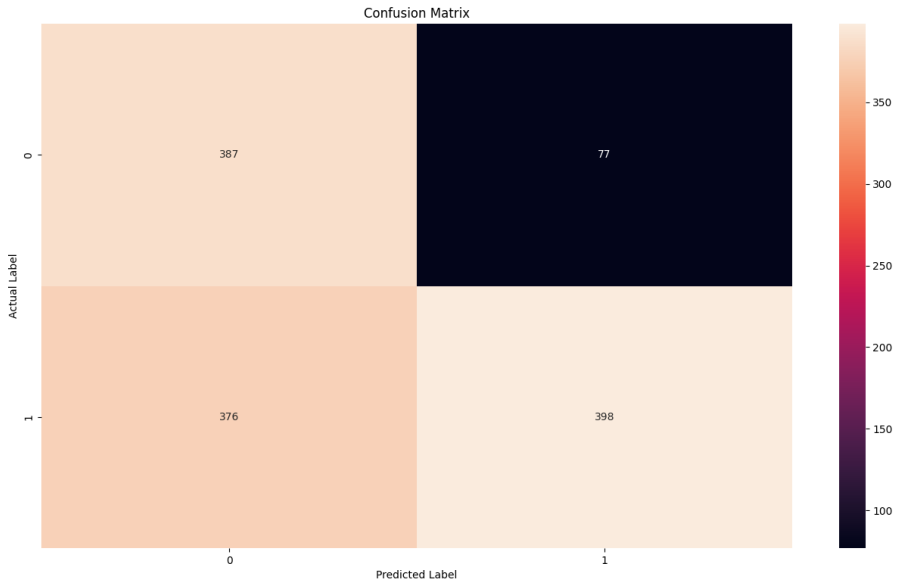
Learning Rate	**Activation Function**	**Layers**	**Neurons**	**Batch Size**	**Patience**	**Max Epochs**
0.1	Relu	3	64	1024	Vypnutý ES	2000



Obr. 20: Vývoj chybovej hodnoty počas trénovania



Obr. 21: Vývoj úspešnosti počas trénovania



Obr. 22: Konfúzna matica (High Learning Rate)

Classification Report:

	precision	recall	f1-score	support
0	0.51	0.83	0.63	464
1	0.84	0.51	0.64	774
accuracy			0.63	1238
macro avg	0.67	0.67	0.63	1238
weighted avg	0.71	0.63	0.63	1238

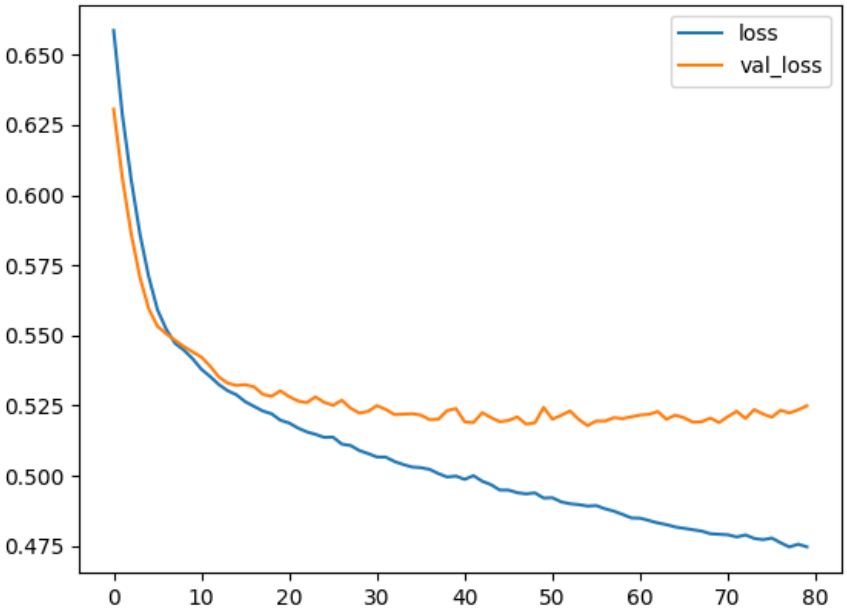
Posledným experimentom je skúmanie siete pri vymazaní nerelevantných stĺpcov. Ako sme už na začiatku spomínali, môžeme zobraziť koreláciu medzi jednotlivými stĺpcami. Takto vyzerá korelácia stĺpcu Quality s ostatnými stĺpcami.

```
density              -0.276095
volatile acidity     -0.258214
chlorides            -0.180525
type                 -0.109787
fixed acidity        -0.060167
total sulfur dioxide -0.053554
residual sugar       -0.041768
pH                    0.020164
free sulfur dioxide   0.040971
sulphates             0.045478
citric acid           0.094210
alcohol               0.414508
quality               1.000000
```

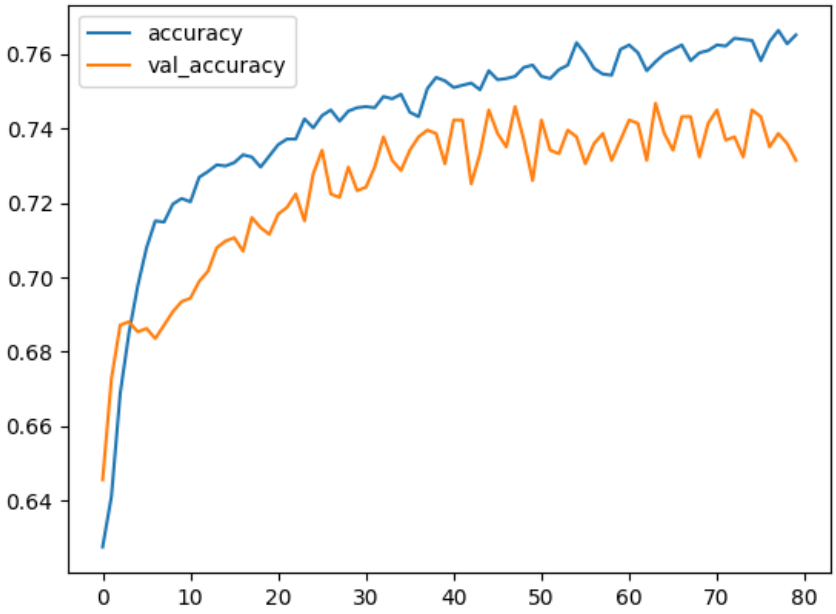
Pre tieto experimenty budeme používať našu najúspešnejšiu sieť:

Learning Rate	Activation Function	Layers	Neurons	Batch Size	Patience
0.001	Relu	3	64	1024	25

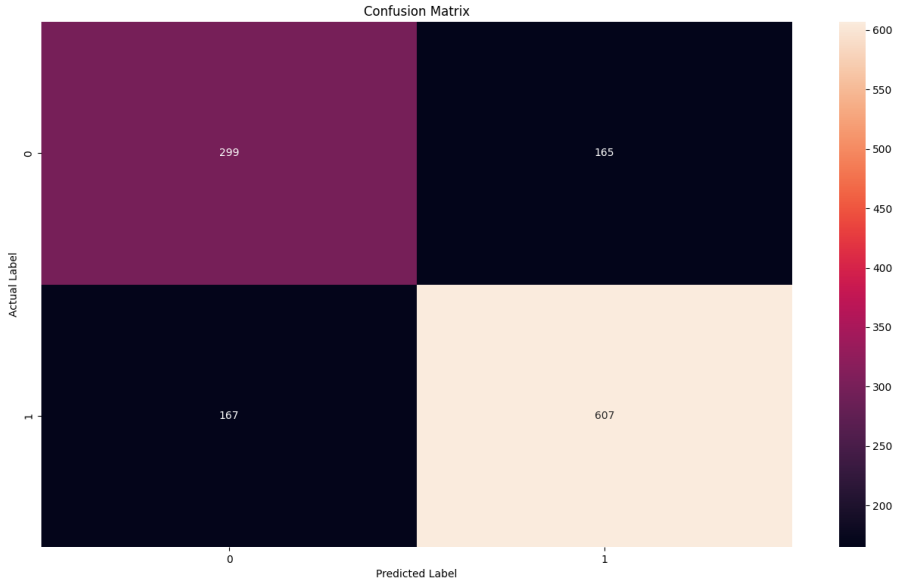
Najprv si vymažeme stĺpce s negatívnou koreláciou. Dosiahli sme 73 percentnú úspešnosť. Early Stopping bol zavolaný po 80. epoche.



Obr. 23: Vývoj chybovej hodnoty počas trénovania



Obr. 24: Vývoj úspešnosti počas trénovania

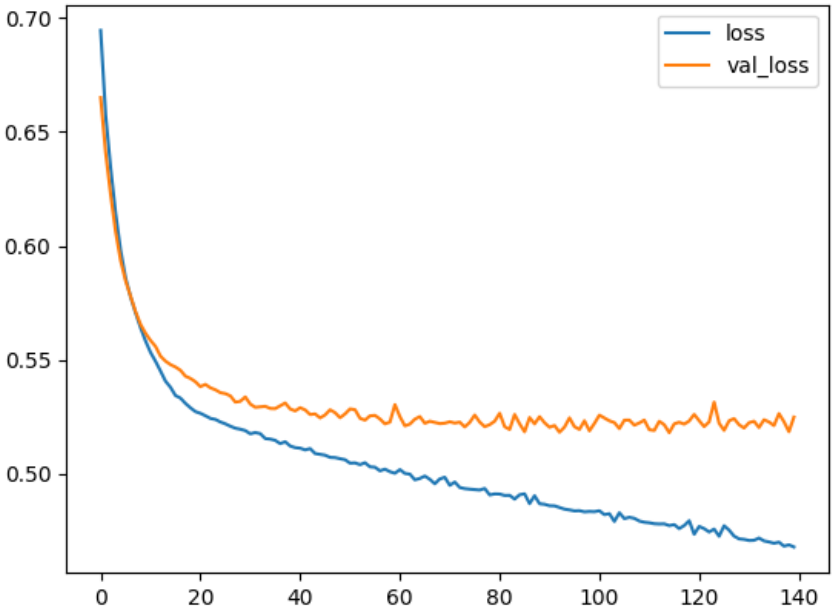


Obr. 25: Konfúzna matica (Vymazaná Negatívna Korelácia)

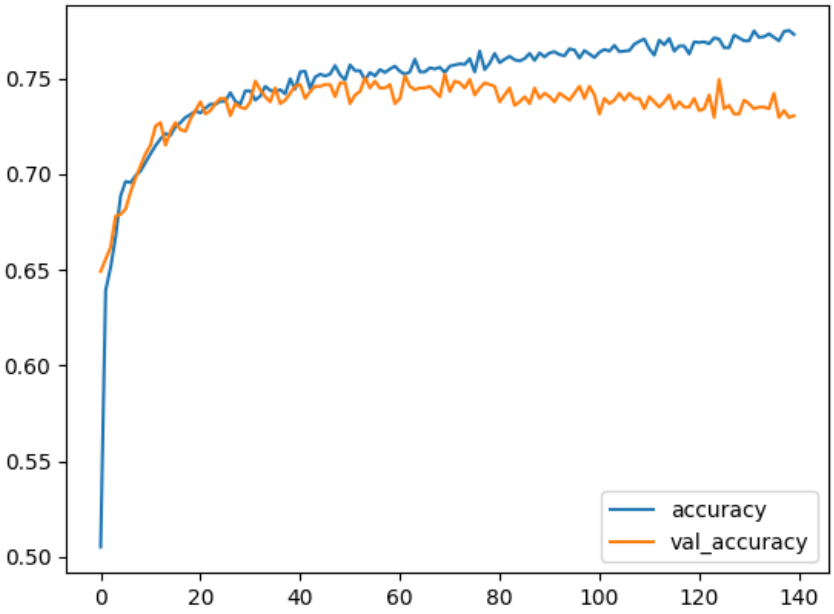
Classification Report:

	precision	recall	f1-score	support
0	0.64	0.64	0.64	464
1	0.79	0.78	0.79	774
accuracy			0.73	1238
macro avg	0.71	0.71	0.71	1238
weighted avg	0.73	0.73	0.73	1238

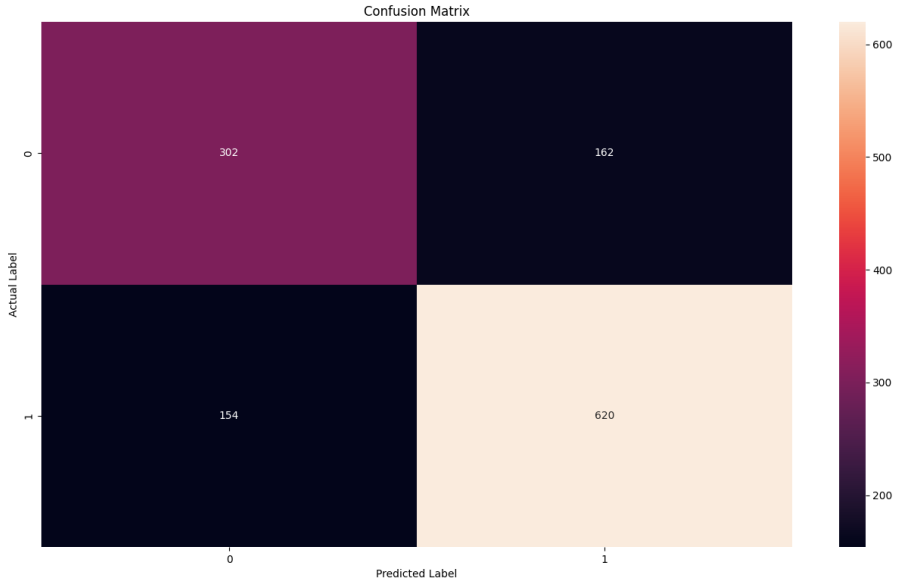
Nakoniec si vymažeme stĺpce s pozitívnou koreláciou. Dosiahli sme 74 percentnú úspešnosť. Early Stopping bol zavolaný po 140. epoche, takže trénovanie bez pozitívne korelovaných dát trvá trochu dlhšie.



Obr. 26: Vývoj chybovej hodnoty počas trénovania



Obr. 27: Vývoj úspešnosti počas trénovania



Obr. 28: Konfúzna matica (Vymazaná Pozitívna Korelácia)

Classification Report:

	precision	recall	f1-score	support
0	0.66	0.65	0.66	464
1	0.79	0.80	0.80	774
accuracy			0.74	1238
macro avg	0.73	0.73	0.73	1238
weighted avg	0.74	0.74	0.74	1238

Zoznam použitej literatúry

1. *pandas* [online] [cit. 2021-10-19]. Dostupné z: https://pandas.pydata.org/.
2. *Matplotlib* [online] [cit. 2021-10-19]. Dostupné z: https://matplotlib.org/.
3. *Seaborn - Statistical Data Visualization* [online] [cit. 2021-10-19]. Dostupné z: https://seaborn.pydata.org/.
4. *scikit-learn* [online] [cit. 2021-10-19]. Dostupné z: https://scikit-learn.org/stable/.
5. *Tensorflow* [online] [cit. 2021-10-19]. Dostupné z: https://www.tensorflow.org/.
6. *Keras - Simple. Flexible. Powerful.* Dostupné tiež z: https://keras.io/.
7. *pandas.read_csv* [online] [cit. 2021-10-19]. Dostupné z: https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html.
8. *pandas.DataFrame.head* [online] [cit. 2021-10-19]. Dostupné z: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.head.html.
9. *pandas.DataFrame.info* [online] [cit. 2021-10-19]. Dostupné z: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.info.html.
10. *pandas.DataFrame.describe* [online] [cit. 2021-10-19]. Dostupné z: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.describe.html.
11. *pandas.DataFrame.transpose* [online] [cit. 2021-10-19]. Dostupné z: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.transpose.html.
12. *StandardScaler* [online] [cit. 2021-10-26]. Dostupné z: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.
13. BROWNLEE, Jason. *Data Leakage in Machine Learning* [online]. 2020-08-15 [cit. 2021-10-19]. Dostupné z: https://machinelearningmastery.com/data-leakage-machine-learning/.
14. *Sklearn.model_selection.GridSearchCV* [online] [cit. 2021-10-26]. Dostupné z: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.
15. GODOY, Daniel. *Understanding binary cross-entropy / log loss: A visual explanation* [online]. Towards Data Science, 2018-11-21 [cit. 2021-10-26]. Dostupné z: https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a.

16. *Sigmoid function* [online]. Wikimedia Foundation [cit. 2021-10-26]. Dostupné z: https://en.wikipedia.org/wiki/Sigmoid_function.

17. *Keras Documentation: EarlyStopping* [online] [cit. 2021-10-26]. Dostupné z: https://keras.io/api/callbacks/early_stopping/.

18. KARAGIANNAKOS, Sergios. *Best practices to write Deep learning code: Project structure, OOP, type checking and documentation* [online]. Sergios Karagiannakos, 2020-06-17 [cit. 2021-10-26]. Dostupné z: https://theaisummer.com/best-practices-deep-learning-code/.

Prílohy

A Štruktúra projektu

Inšpiráciu pre projektovú štruktúru sme našli na webovej stránke AI Summer [18].

configs

· Konfiguračné súbory

/config.py

· Hlavný konfiguračný súbor

/units.py

· Jednotky pre jednotlivé stĺpce

data

· Dátové súbory

/wine_test.csv

· Testovacie dáta

/wine_train.csv

· Trénovacie dáta

dataloader

· Čítač dát

/dataloader.py

· Čítač dát

neural_network_project

· Spúšťač

/neural_network_project.py

· Spúšťač zadania

models

· Modely Strojového Učenia

/base_model.py

· Abstraktný Model

/neural_network.py

· Neurónová Sieť

ops

· Operácie

/plotter.py

· Vykresľovač grafov

output

· Výstupy

/grid_search_output.txt

· Výstup funkcie Grid Search

utils

· Utilitné funkcie

/setup.py

· Setup metódy

/I-SUNS_-_Neural_Network_Project.pdf

· Dokumentácia - tento dokumnet

/main.py

· Hlavný program

/Neural_Network_Project

· Bash Script

/Neural_Network_Project.ps1

· PowerShell Script

/requirements.txt

· Zoznam požiadnaých balíčkov

B Používateľská príručka

V tejto časti práce prejdeme spôsoby, ktoré nám umožňujú spúšťať túto aplikáciu. Treba špecifikovať, ktorý skript chceme spustiť na základe operačného systému. Potom si musíme špecifikovať v ktorom móde to chceme spúšťať:

- –**best** - Najlepšia konfigurácia
- –**under_train** - Podtrénovacia konfigurácia
- –**over_train** - Pretrénovacia konfigurácia
- –**fast_train** - Konfigurácia so zrýchleným učením
- –**slow_train** - Konfigurácia so spomaleným učením

Linux

```
$ ./Neural_Network_Project [ --best | --under_train | --over_train |
                             --fast_train | --slow_train ]
```

Windows

```
> .\Neural_Network_Project.ps1 [ --best | --under_train | --over_train |
                                 --fast_train | --slow_train ]
```