



**RĪGAS TEHNISKĀ UNIVERSITĀTE**  
**Datorzinātnes un informācijas tehnoloģijas fakultāte**  
**Informācijas tehnoloģijas institūts**

**Atskaite**  
**mācību priekšmetā**  
**“Mākslīgā intelekta pamati”**

**“Mašīnmācīšana”**

Izstrādāja:  
Andrejs Golovanovs  
DITF 2.kurss, 8.grupa  
St.Apl.nr.: 191RKB058

2021./22. māc. gads

## Saturs

<i>Datu kopas apraksts</i> .....	4
<i>Datu kopas satura apraksts</i> .....	4
<i>Secinājumi(I)</i> .....	5
<i>Nepārraudzītā mašīnmācīšanās</i> .....	8
<i>Hierarhiskā klasterizācija</i> .....	8
<i>Secinājumi (II)</i> .....	10

## Datu kopas apraksts

Savā darbā es izmantoju datu kopu, kas ir zemāk esošajā saitē.

<https://www.kaggle.com/datasets/brsdincer/star-type-classification>

Datu kopas nosaukums "Zvaigžņu tipu klasifikācija". Vietnē augšupielādēja Baris Dinser, bet izveidotājs – NASA. Dati tika iegūti, lai salīdzinātu visus ML modeļus, bet tos var izmantot prognozēšanai.

Licencēšanas nosacījumi:

<https://opendatacommons.org/licenses/dbcl/1-0/>

## Datu kopas satura apraksts

Tabulā ir 240 rindas. Tabulā bija 2 klases (Color un Spectral\_Class), bet vienu ciparu klasi nomainīju uz tekstu, kas pēc būtības neko neietekmēja (Type). Tabulā ir 7 kolonnas.

1	Temperat	L	R	A_M	Color	Spectral_C	Type
2	3068	0,002400	0,170000	16,120000	Red	M	Red_Dwarf
3	3042	0,000500	0,154200	16,600000	Red	M	Red_Dwarf
4	2600	0,000300	0,102000	18,700000	Red	M	Red_Dwarf
5	2800	0,000200	0,160000	16,650000	Red	M	Red_Dwarf
6	1939	0,000138	0,103000	20,060000	Red	M	Red_Dwarf
7	2840	0,000650	0,110000	16,980000	Red	M	Red_Dwarf
8	2637	0,000730	0,127000	17,220000	Red	M	Red_Dwarf
9	2600	0,000400	0,096000	17,400000	Red	M	Red_Dwarf
10	2650	0,000690	0,110000	17,450000	Red	M	Red_Dwarf
11	2700	0,000180	0,130000	16,050000	Red	M	Red_Dwarf
12	3600	0,002900	0,510000	10,690000	Red	M	Brown_Dwarf
13	3129	0,012200	0,376100	11,790000	Red	M	Brown_Dwarf
14	3134	0,000400	0,196000	13,210000	Red	M	Brown_Dwarf
15	3628	0,005500	0,393000	10,480000	Red	M	Brown_Dwarf
16	2650	0,000600	0,140000	11,782000	Red	M	Brown_Dwarf
17	3340	0,003800	0,240000	13,070000	Red	M	Brown_Dwarf
18	2799	0,001800	0,160000	14,790000	Red	M	Brown_Dwarf
19	3692	0,003670	0,470000	10,800000	Red	M	Brown_Dwarf
20	3192	0,003620	0,196700	13,530000	Red	M	Brown_Dwarf
21	3441	0,039000	0,351000	11,180000	Red	M	Brown_Dwarf
22	25000	0,056000	0,008400	10,580000	Blue Whit	B	White_Dwarf
23	7740	0,000490	0,012340	14,020000	White	A	White_Dwarf
24	7220	0,000170	0,011000	14,230000	White	F	White_Dwarf

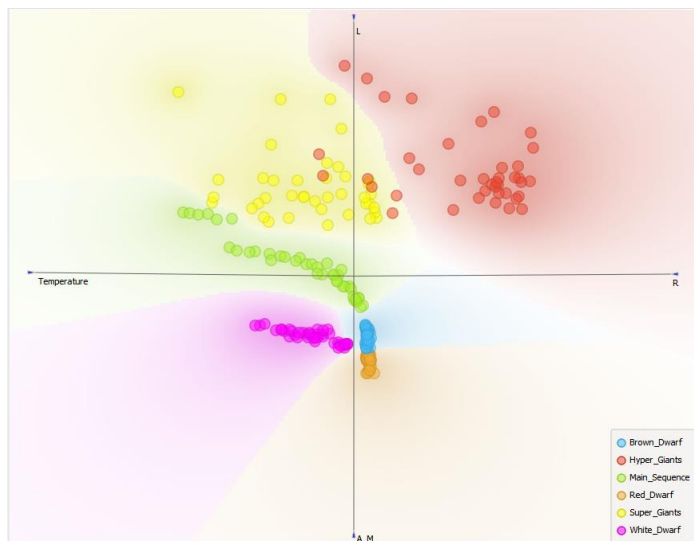
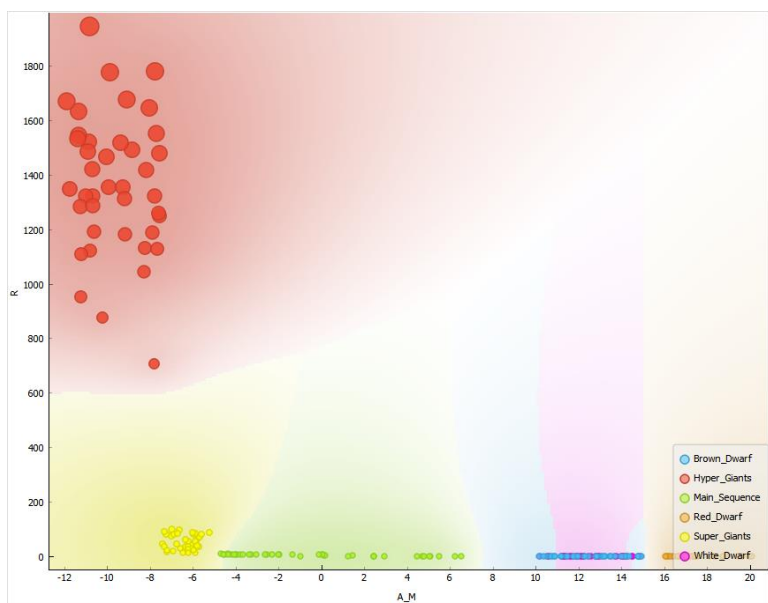
Att.1

Temperature -- K  
 L -- L/Lo - Spilgtums  
 R -- R/Ro - Radius  
 AM – Mv - Absolūtais spožums  
 Color -- General Color of Spectrum  
 Spectral\_Class -- O,B,A,F,G,K,M / SMASS - [https://en.wikipedia.org/wiki/Asteroid\\_spectral\\_types](https://en.wikipedia.org/wiki/Asteroid_spectral_types)  
 Type -- Red Dwarf, Brown Dwarf, White Dwarf, Main Sequence, Super Giants, Hyper Giants

$L_o = 3.828 \times 10^{26}$  Watts  
 (Avg Luminosity of Sun)  
 $R_o = 6.9551 \times 10^8$  m  
 (Avg Radius of Sun)

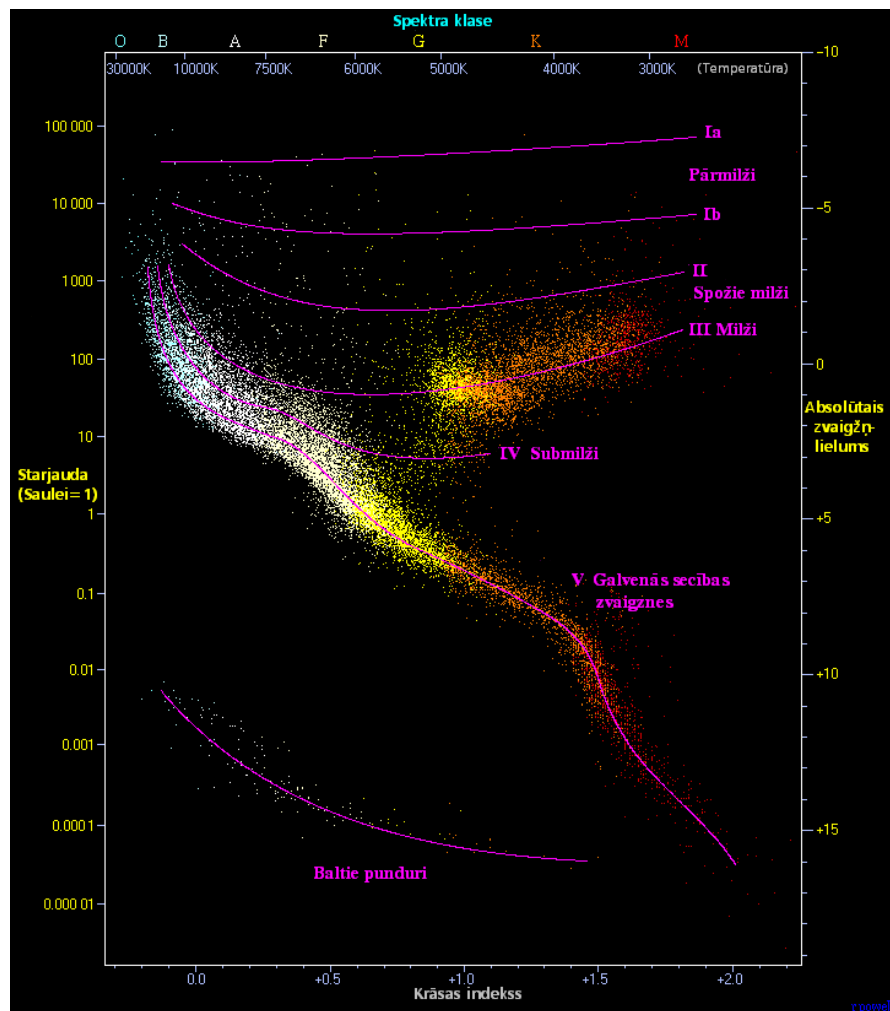
## Secinājumi(I)

Strādājot ar šo tabulu, pamanīju, ka dominē viena klase - Tips. Ja mēģināt kārtot datus pēc citām klasēm, daudzi no šiem datiem ir sajaukti. Un tad dominējošās klases objekti ne visur ir labi sakārtoti. hipergianti ir nedaudz sajaukti ar supergigantiem, izmantojot 3-dimensiju vizualizāciju, savukārt 2-dimensiju vizualizācijā brūnie punduri un baltie punduri sajaukti kopā.



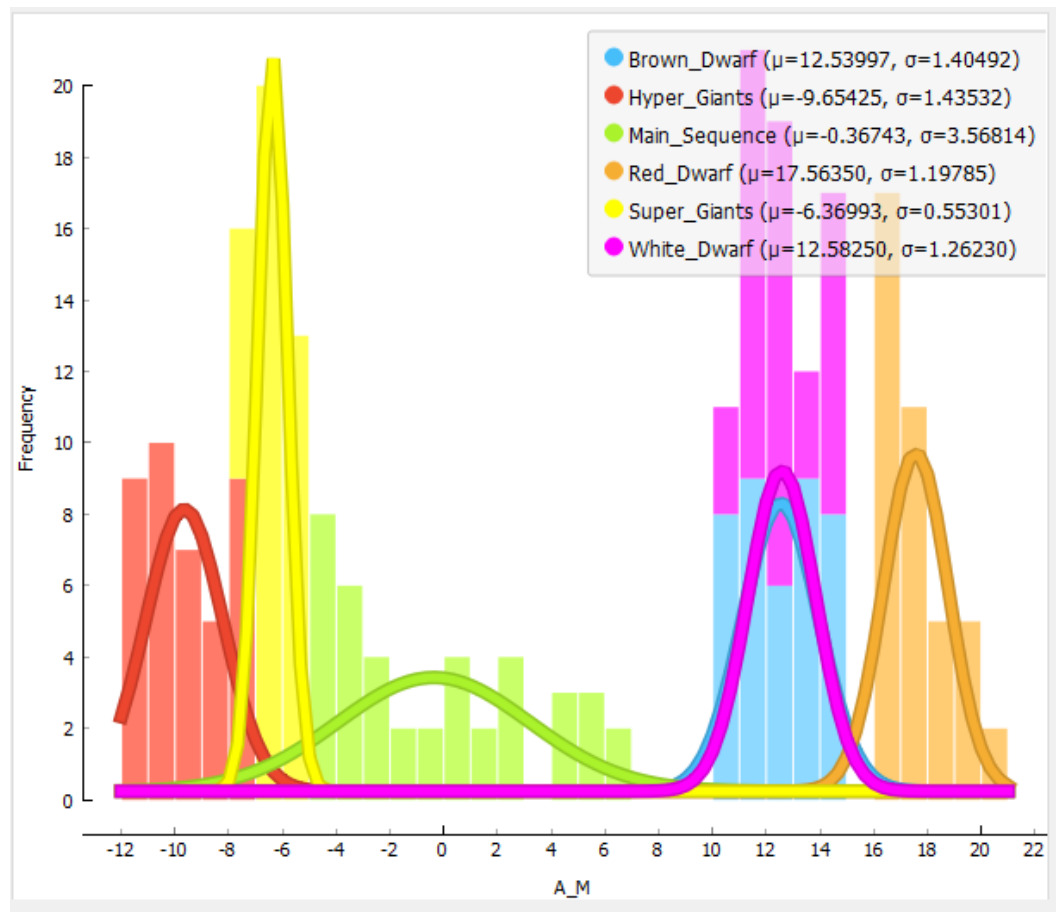
Att.2 un Att.3

Salīdzinot ar Herčšprungas-Raselas diagrammu, varam secināt, ka zvaigznes nevar perfekti atdalīt, jo dažām zvaigznēm var būt īpašības, kas tuvas vieniem un tiem pašiem supergigantiem, bet būtība ir hipergianti.

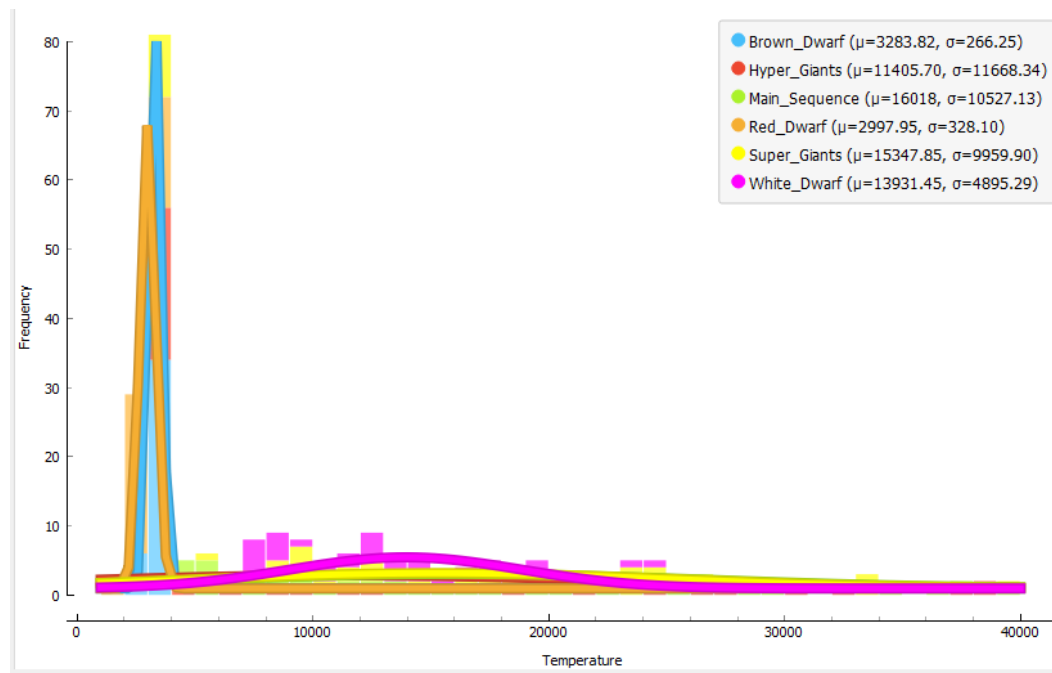


Att.4

Aplūkojot standartnovirzi, kas nepārsniedz 3,5, varu secināt, ka izkliede attiecībā pret absolūto lielumu nav īpaši liela, kad pārējās histogrammas vērtība pārsniedz vairākus tūkstošus, minēšu temperatūras piemēru. Tas liek domāt, ka izkliede ir liela, taču tādas ir arī pašas temperatūras vērtības, tāpēc šāda izkliede ir diezgan sagaidāma.



Att.5

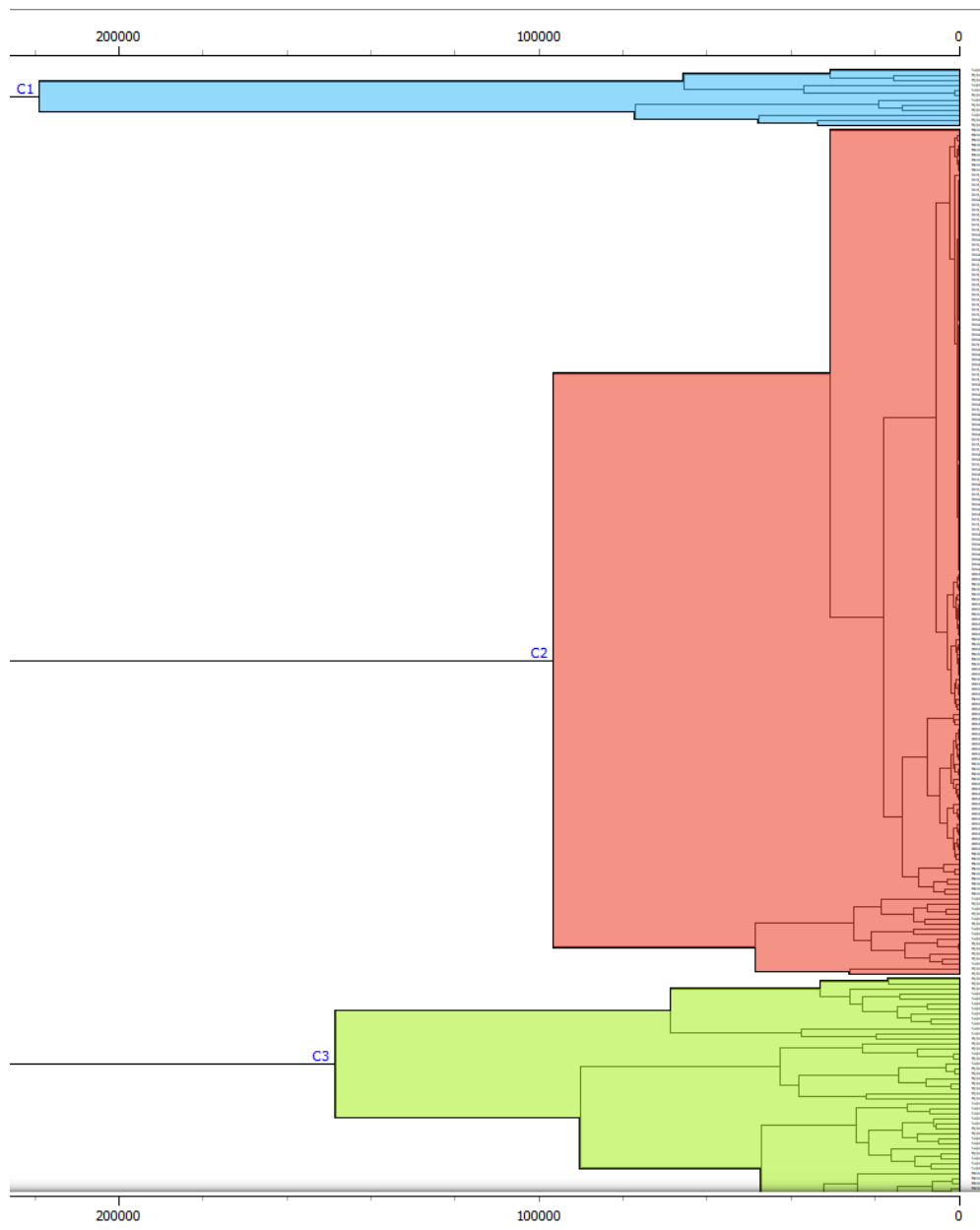


Att.6

# Nepārraudzītā mašīnmācīšanās

## Hierarhiskā klasterizācija

Es eksperimentēju ar hierarhisku klasterizāciju 3 reizes, nejauši mainot hiperparametrus. Es iestatīju dziļumu uz 100, ievietoju svērtu saiti (Linkage: Weighted)) un pēc tam mainīju kopu skaitu. Sākotnēji tas bija 3, un es nolēmu to uzņemt kā pirmo eksperimentu. Manuprāt, viņi atdalījās diezgan labi, pirmajā klasterī bija milži, otrajā punduri un galvenā virkne, un arī trešajā parādījās milži.

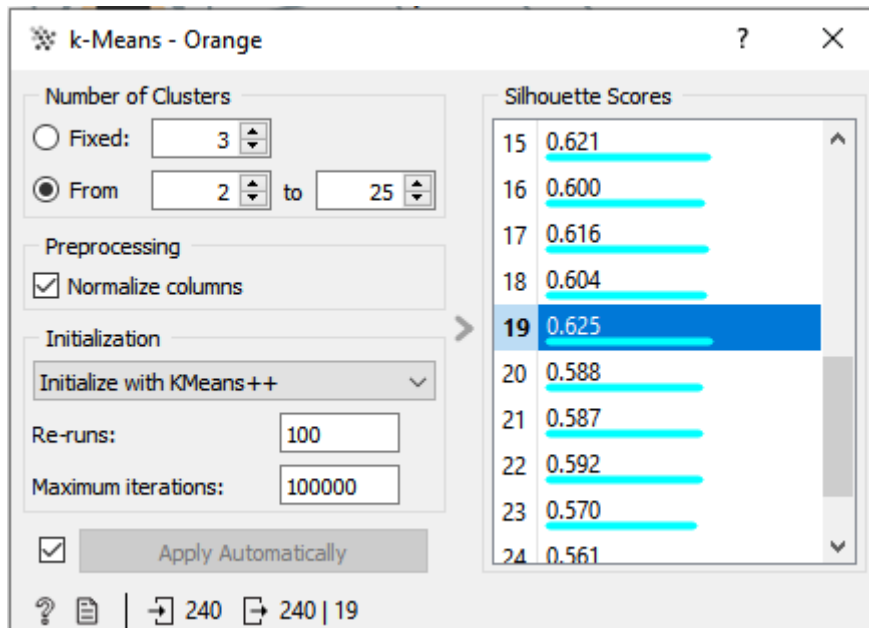


Att.7

Es tur veicu daudz eksperimentu, bet, sasniedzot maksimālo vērtību (20), es nevarēju atdalīt pundurus vienu no otra un galveno secību. Tika nolemts pāriet uz slīdņi. Kad augstums bija 0,4%, es varēju atdalīt galveno secību no punduriem, un klasteru skaits sasniedza 90.

## K-vidējo algoritms

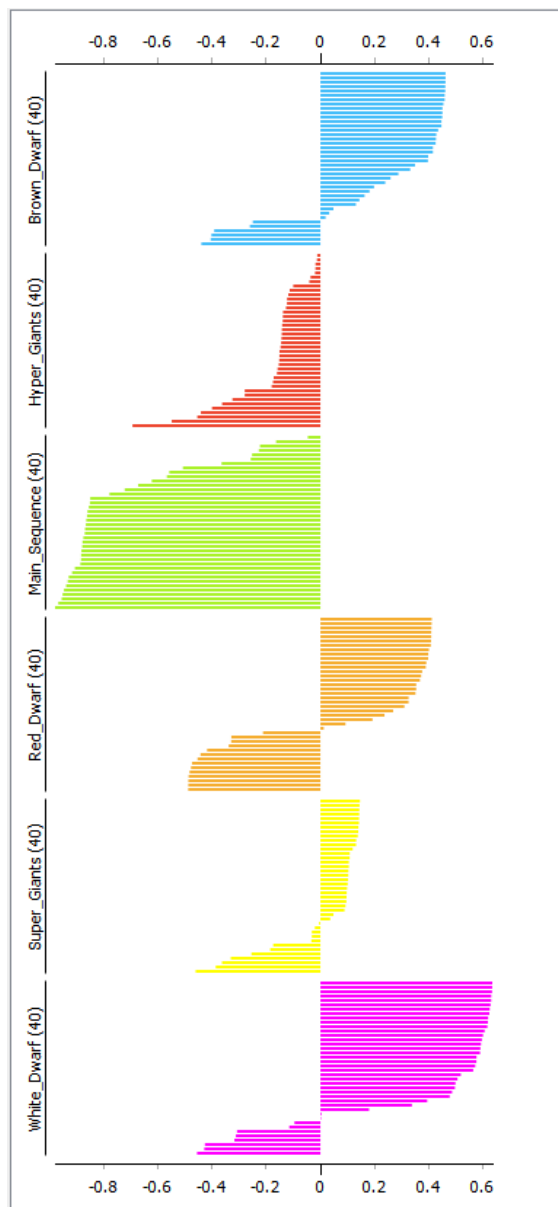
Es izmantoju K-means algoritmu un nolēmu palaist klasteru skaitu no 2 līdz 25. Izrādījās, ka labākā vērtība izrādījās 19 klasteri, bet tā vērtība izrādījās tikai 0,625, kas nozīmē, ka gandrīz puse būs slikti sakārtotam.



Att.8

Un tā arī notika. Gandrīz puse no visām zvaigznēm atrodas negatīvo vērtību zonā, galvena secība un hipergigantie pilnīgi atrodas negatīvo vērtību zonā, kas nozīmē, ka tās ir grūti izšķirt.





Att.9

## Secinājumi (II)

Pēc daudziem eksperimentiem es sapratu, ka ir grūti atdalīt zvaigžņu grupas vienu no otras. Tas ir saistīts ar faktu, ka daudzām zvaigznēm ir līdzīgi parametri, piemēram, tiem pašiem punduriem un galvenās secības zvaigznēm ir līdzīgi parametri. Un hipergianti ir ļoti līdzīgi supergiantiem. Lai tos labāk attālinātu vienu no otra, ideāli būtu pievienot vairāk parametru, piemēram, masu.

## Pārraudzītā mašīnmācīšanās

Es izvēlējos divus algoritmus: kNN un Loģistiska regresija. Es izvēlējos šos divus algoritmus, jo kNN ir diezgan vienkāršs un Logistic Regression sniedz izcilus rezultātus salīdzinājumā ar citiem algoritmiem.

kNN-metriskais algoritms objektu automātiskai klasifikācijai vai regresijai.

Klasifikācijas metodes izmantošanas gadījumā objekts tiek piešķirts klasei, kas ir visizplatītākā starp šī elementa K kaimiņiem, kuru klases jau ir zināmas. Gadījumā, ja tiek izmantota regresijas metode, objektam tiek piešķirta tam tuvāko K objektu vidējā vērtība, kuru vērtības jau ir zināmas.

[https://en.wikipedia.org/wiki/K-nearest\\_neighbors\\_algorithm](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm)

Loģistisko regresiju izmanto, lai prognozētu notikuma iespējamību, pamatojoties uz pazīmju kopas vērtībām. Šim nolūkam tiek ieviests tā sauktais atkarīgais mainīgais Y, kam ir tikai viena no divām vērtībām.

[https://en.wikipedia.org/wiki/Logistic\\_regression](https://en.wikipedia.org/wiki/Logistic_regression)

## Neironu tīkli

Sāku ar neironu tīkliem. Viņiem ir hiperparametrs - neironu skaits slēptajā slānī. Pēc noklusējuma ir 1 slānis ar 100 neironiem. Ar šiem parametriem viņa CA ir 0,976(Att.10), kas ir lielisks rezultāts. Es nolēmu pievienot vēl vienu slāni ar 50 neironiem, un rezultāts bija 0,988(Att.11). Trešajā eksperimentā es palielināju skaitli otrajā slānī līdz 100, un rezultāts bija 0,992(Att.12). Mēģinot pievienot jaunus slāņus, AC īpaši nemainījās, tikai devās zemāk.

		Predicted						Σ
		Brown_Dwarf	Hyper_Giants	Main_Sequence	Red_Dwarf	Super_Giants	White_Dwarf	
Actual	Brown_Dwarf	274	0	0	3	0	0	277
	Hyper_Giants	0	265	2	0	1	0	268
	Main_Sequence	0	0	265	0	10	0	275
	Red_Dwarf	0	0	0	277	0	0	277
	Super_Giants	0	0	4	0	268	0	272
	White_Dwarf	0	0	0	0	0	271	271
Σ		274	265	271	280	279	271	1640

		Predicted						Σ
		Brown_Dwarf	Hyper_Giants	Main_Sequence	Red_Dwarf	Super_Giants	White_Dwarf	
Actual	Brown_Dwarf	256	0	0	21	0	0	277
	Hyper_Giants	0	267	0	0	1	0	268
	Main_Sequence	0	0	269	0	6	0	275
	Red_Dwarf	0	0	0	277	0	0	277
	Super_Giants	0	0	11	0	261	0	272
	White_Dwarf	0	0	0	0	0	271	271
Σ		256	267	280	298	268	271	1640

		Predicted						Σ
		Brown_Dwarf	Hyper_Giants	Main_Sequence	Red_Dwarf	Super_Giants	White_Dwarf	
Actual	Brown_Dwarf	276	0	0	1	0	0	277
	Hyper_Giants	0	268	0	0	0	0	268
	Main_Sequence	0	0	265	0	10	0	275
	Red_Dwarf	0	0	0	277	0	0	277
	Super_Giants	0	0	2	0	270	0	272
	White_Dwarf	0	0	0	0	0	271	271
Σ		276	268	267	278	280	271	1640

Att.10, Att.11 un Att.12

## kNN

Pārejot uz kNN, nebija daudz hiperparametru. Bija skaitlis, kas ir atbildīgs par algoritmā ņemto kaimiņu skaitu. Sākotnējais parametrs bija 5. Pārejot uz kNN, nebija daudz hiperparametru. Bija skaitlis, kas ir atbildīgs par algoritmā ņemto kaimiņu skaitu. Sākotnējais parametrs bija 5. Piecos viņa CA bija 0,642(Att.13). Tad es šo parametru padarīju vienādu ar 10. Tā vērtība samazinājās līdz 0,615(Att.14). Tāpēc es nolēmu samazināt parametru līdz 3. Vērtība kļuva par 0,662(Att.15).

		Predicted						Σ
		Brown_Dwarf	Hyper_Giants	Main_Sequence	Red_Dwarf	Super_Giants	White_Dwarf	
Actual	Brown_Dwarf	201	0	0	76	0	0	277
	Hyper_Giants	0	140	2	0	126	0	268
	Main_Sequence	11	3	151	0	4	106	275
	Red_Dwarf	108	0	0	169	0	0	277
	Super_Giants	0	108	10	0	154	0	272
	White_Dwarf	0	0	33	0	0	238	271
Σ		320	251	196	245	284	344	1640

		Predicted						Σ
		Brown_Dwarf	Hyper_Giants	Main_Sequence	Red_Dwarf	Super_Giants	White_Dwarf	
Actual	Brown_Dwarf	234	0	0	43	0	0	277
	Hyper_Giants	0	173	2	0	93	0	268
	Main_Sequence	22	2	117	0	31	103	275
	Red_Dwarf	116	0	0	161	0	0	277
	Super_Giants	0	168	5	0	99	0	272
	White_Dwarf	0	0	46	0	0	225	271
Σ		372	343	170	204	223	328	1640

	Predicted						$\Sigma$
	Brown_Dwarf	Hyper_Giants	Main_Sequence	Red_Dwarf	Super_Giants	White_Dwarf	
Actual	Brown_Dwarf	189	0	0	88	0	277
	Hyper_Giants	0	142	0	0	126	268
	Main_Sequence	8	5	177	0	3	275
	Red_Dwarf	107	0	0	170	0	277
	Super_Giants	0	98	11	0	163	272
	White_Dwarf	0	0	27	0	0	271
	$\Sigma$	304	245	215	258	292	1640

Att.13, Att.14 un Att.15

## Loģistika regresija

Strādājot ar loģistikas regresiju, viņai bija viens parametrs, spēks, kas sākotnēji tika iestatīts uz 10. Viņa CA bija 0,968(Att.16). Pēc tam es iestatīju stipruma parametru uz 1000, kā rezultātā vērtība kļuva par 0,969(Att.17). Pēdējo reizi es iestatīju vērtību uz 0,001. Vērtība kļuva par 0,793(Att.18).

	Predicted						$\Sigma$
	Brown_Dwarf	Hyper_Giants	Main_Sequence	Red_Dwarf	Super_Giants	White_Dwarf	
Actual	Brown_Dwarf	267	0	0	10	0	277
	Hyper_Giants	0	265	1	0	2	268
	Main_Sequence	2	0	255	0	16	275
	Red_Dwarf	11	0	0	266	0	277
	Super_Giants	0	0	9	0	263	272
	White_Dwarf	0	0	0	0	0	271
	$\Sigma$	280	265	265	276	281	1640

		Predicted						Σ
		Brown_Dwarf	Hyper_Giants	Main_Sequence	Red_Dwarf	Super_Giants	White_Dwarf	
Actual	Brown_Dwarf	269	0	0	8	0	0	277
	Hyper_Giants	0	258	1	0	9	0	268
	Main_Sequence	0	0	258	0	15	2	275
	Red_Dwarf	6	0	0	271	0	0	277
	Super_Giants	0	1	9	0	262	0	272
	White_Dwarf	0	0	0	0	0	271	271
Σ		275	259	268	279	286	273	1640

		Predicted						Σ
		Brown_Dwarf	Hyper_Giants	Main_Sequence	Red_Dwarf	Super_Giants	White_Dwarf	
Actual	Brown_Dwarf	58	0	0	214	0	5	277
	Hyper_Giants	0	267	0	0	1	0	268
	Main_Sequence	0	0	167	0	15	93	275
	Red_Dwarf	0	0	0	277	0	0	277
	Super_Giants	0	0	11	0	261	0	272
	White_Dwarf	0	0	0	0	0	271	271
Σ		58	267	178	491	277	369	1640

Att.16, Att.17 un Att.18

Uztrenējis mākslīgo intelektu, nolēmu to pārbaudīt, izveidojot testa tabulu

Temperat	L	R	A_M	Color	Spectral_Type
39000	204000	10	-7	-	O ?
15000	150000	2000	-10	-	B ?
8000	0,0004	0,012	14	-	A ?
4000	0,01	0,15	17	-	M ?
10000	9	2	-1	-	F ?
5000	500000	1200	-8	-	K ?
10000	20	2008	4	-	K ?

Att.19

Uzdevums bija noteikt, kāda veida ir šīs zvaigznes. Diemžēl man nebija pareizās atbildes, jo šīs zvaigznes tika izdomātas no manas galvas, bet visas zvaigznes, izņemot pēdējo sarakstā, bija pēc sākotnējās tabulas. Pēdējo izdomāja mans brālis, un viņš ieviesa vērtības, kuras gribēja, lai tikai paspēlētos.

	Neural Network	kNN	Logistic Regression	Temperature	L	R	A_M	Spectral_
1	Main_Sequence	Main_Seque...	Main_Sequence	39000	204000.0000	10.000	-7	O
2	Hyper_Giants	Super_Giants	Hyper_Giants	15000	150000.0000	2000.000	-10	B
3	White_Dwarf	Main_Seque...	White_Dwarf	8000	0.0004	0.012	14	A
4	Red_Dwarf	Brown_Dwarf	Red_Dwarf	4000	0.0100	0.150	17	M
5	Main_Sequence	Main_Seque...	White_Dwarf	10000	9.0000	2.000	-1	F
6	Hyper_Giants	Super_Giants	Hyper_Giants	5000	500000.0000	1200.000	-8	K
7	Hyper_Giants	Main_Seque...	Hyper_Giants	10000	20.0000	2008.000	4	K

Att.20

### Secinājumi (III)

Salīdzinot trīs algoritmu darbību, varu secināt, ka neironu tīkls sevi parādīja vislabāk no visiem. Viņa, visticamāk, pareizi nosauca zvaigžņu veidus, savukārt pārējām bija zemāks koeficients. Testa tabulā mākslīgais intelekts pareizi norādīja visas zvaigznes, jo datus ņēmu tā nosauktajā tipu diapazonā.