МАТЕМАТИЧКИ ФАКУЛТЕТ У БЕОГРАДУ

Анализа скупа података о леукемији

Семинарски рад из предмета истраживање података 1

Величковић Бојан

Увод

У овом раду бавићемо се скупом података о леукемији преузетим са ове адресе. Скуп података садржи информације о 5 различитих типова леукемије и генској експресији великог броја гена. Генска експресија је мера која се користи за исказивwје коју количину протеина прави дати ген. Тип података те мере је број у децималном запису и што је он већи, то је дати ген експресивнији.

На овом скупу података приказаћемо рад неколико метода за класификацију, кластеровање и на крају ћемо одредити и правила придруживања.

Експлоративна анализа података

Пре бављења икаквим моделима, потребно је прво детаљно анализирати и упознати се са подацима на којима ћемо радити.
Скуп података се састоји од 64 инстанце и 22284 атрибута. Један атрибут је један од пет типова леукемије, а сви остали су експресије разних гена. Одмах на почетку видимо да наш скуп података има релативно мали број инстанци и велики број атрибута.

Типови леукемије који су забележени у скупу су:

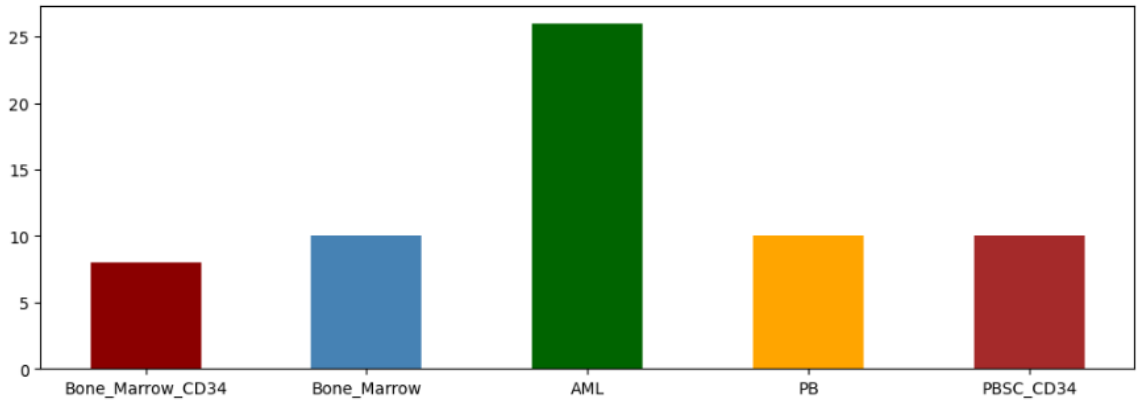
1. АМЛ
2. ПБ
3. ПБ–ЦД34
4. Коштана срж
5. Коштана срж-ЦД34

Овај скуп је преобиман да би могао ручно да се истражује, али срећом, баш над оваквим скуповима коришћење модела за истраживање података има и највише смисла.

	samples	type	1007_s_at	1053_at	117_at	121_at	1255_g_at	1294_at	1316_at	1320_at	...
0	1	Bone_Marrow_CD34	7.745245	7.811210	6.477916	8.841506	4.546941	7.957714	5.344999	4.673364	...
1	12	Bone_Marrow_CD34	8.087252	7.240673	8.584648	8.983571	4.548934	8.011652	5.579647	4.828184	...
2	13	Bone_Marrow_CD34	7.792056	7.549368	11.053504	8.909703	4.549328	8.237099	5.406489	4.615572	...
3	14	Bone_Marrow_CD34	7.767265	7.094460	11.816433	8.994654	4.697018	8.283412	5.582195	4.903684	...
4	15	Bone_Marrow_CD34	8.010117	7.405281	6.656049	9.050682	4.514986	8.377046	5.493713	4.860754	...

Слика 1: Приказ података

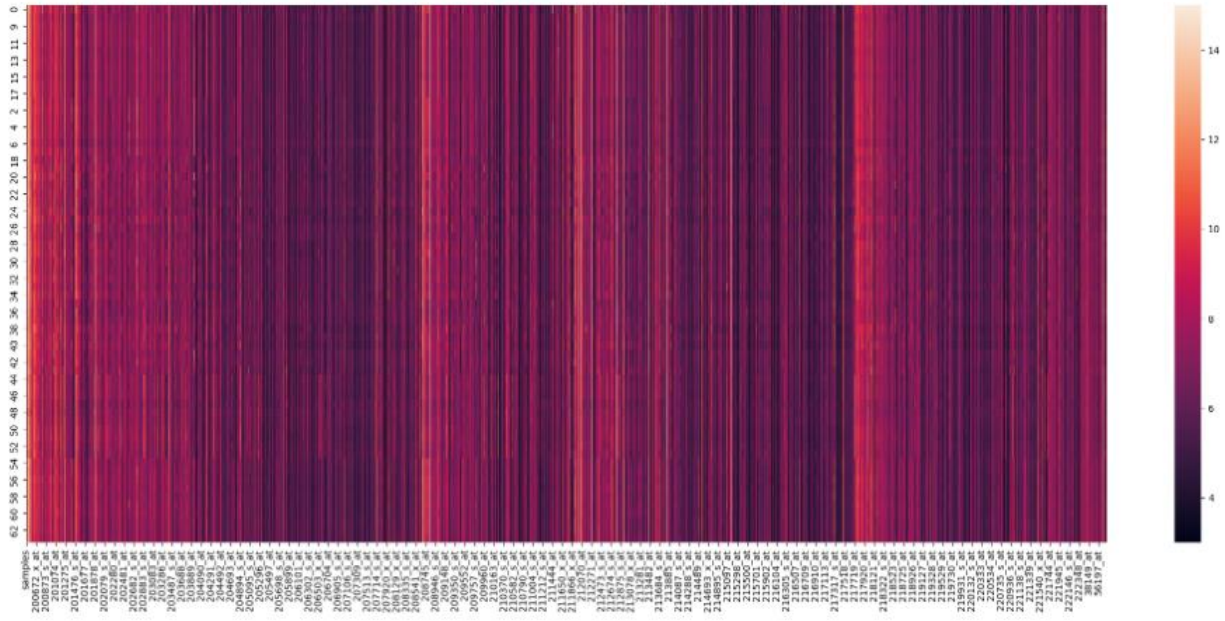
Исцртавањем количине примера сваког типа леукемије у табели добијамо следећи график.



Слика 2: Број примера сваког типа леукемије(8-10-26-10-10)

Са ове слике можемо видети да су наши подаци благо небалансирани јер је број примерака АМЛ леукемије већи од осталих (12,5% - 15,625% - 40,625% - 15,625% - 15,625%). Ова небалансираност није велика, па нема потребе да користимо икакве методе балансирања за сада.

Као што је већ речено, због великог броја атрибута, јако је тешко уочити правилности у овим подацима без помоћи неких алгоритама који ће то урадити за нас. Једна од ствари које можемо урадити је да направимо график зависности инстанци и експресија.



Слика 3: Хит мапа експресије гена по инстанцама

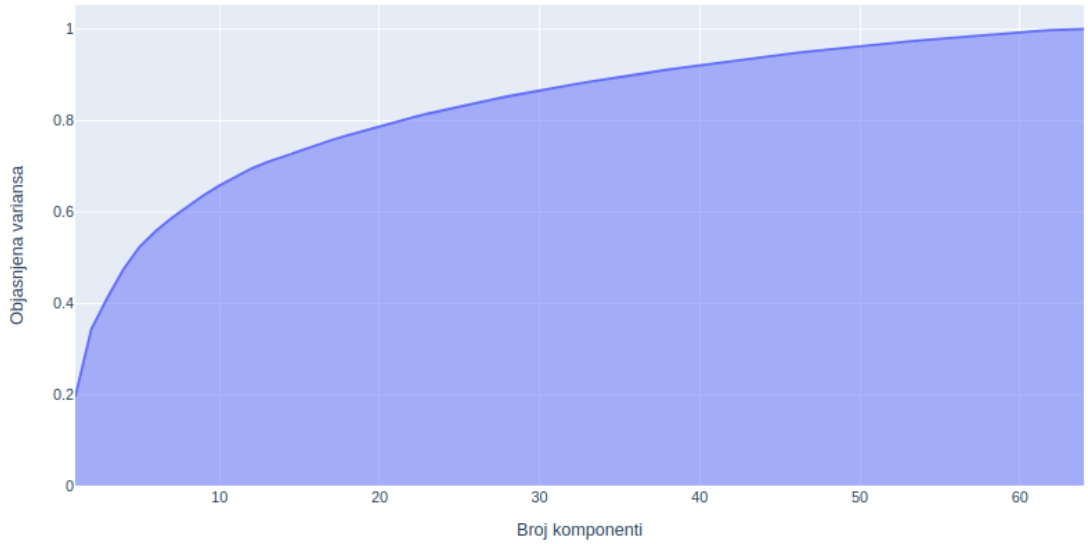
На овом графику, што је ген у датој инстанци експресивнији, то је боја светлија.
Са овог графика можемо закључити да постоји правилност у испољавању гена у виду природно веће експресије неких гена у односу на друге. То закључујемо због тога што су усправне линије на графику јасно видљиве, тј неки гени су експресивнији на свим инстанцама, а неки су мање експресивни.

Можемо и исцртати график на ком ћемо уцртати сваку инстанцу из нашег скупа података и инстанце исте леукемије бојити истом бојом. Овде се јавља проблем са бројем атрибута у нашем скупу.
Да би овај график био 100% исправан, он би морао да има исти број димензија колики је и број атрибута у скупу. Смањењем димензија овог графика губимо и проценат објашњене варијансе, и што је број димензија мањи то нам објашњена варијанса брже опада.

Користићемо ПЦА алгоритам како би смањили димензионалност нашег скупа на два и тада ће нам објашњена варијанса пасти на 34,22%.

Слика 4: График наших инстанци у две димензије

Ако би желели да објашњена варијанса буде преко 65%, наш график би морао да има бар 10 димензија, за преко 80% би биле потребне 22 димензије.



Слика 5: Зависност објашњене варијансе и броја компоненти ПЦА

Претпроцесирање података

Следећи корак би био припрема података за обраду. Испитивањем података сазнајемо да наш скуп не садржи ни недостајуће, а ни нула вредности, па овим не морамо више да се бавимо.

Сада морамо да се бавимо аутлајерима.
Као што је већ речено, неки гени су природно експресивнији у свим инстанцама, тј сваки ген има свој интервал у који можемо сместити његове експресије за све инстанце. Притом, не постоји ни један ген, ком је разлика максималне и минималне експресије већа од 9, што нам даје за право да са великом сигурношћу тврдимо да није долазило до грешака приликом уношења података. Другим речима, сви аутлајери који се јављају у табели су се природно јавили, и као такви, њихово појављивање само по себи носи битне информације. Ако су баш ти аутлајери битни за одређивање типа леукемије, онда не желимо да их избацујемо из наше табеле, а са друге стране, ако они нису ни у ком смислу битни, онда чак и ако се налазе у табели, неће изазивати проблеме. Дакле, за сада нећемо избацивати ништа из нашег скупа података.

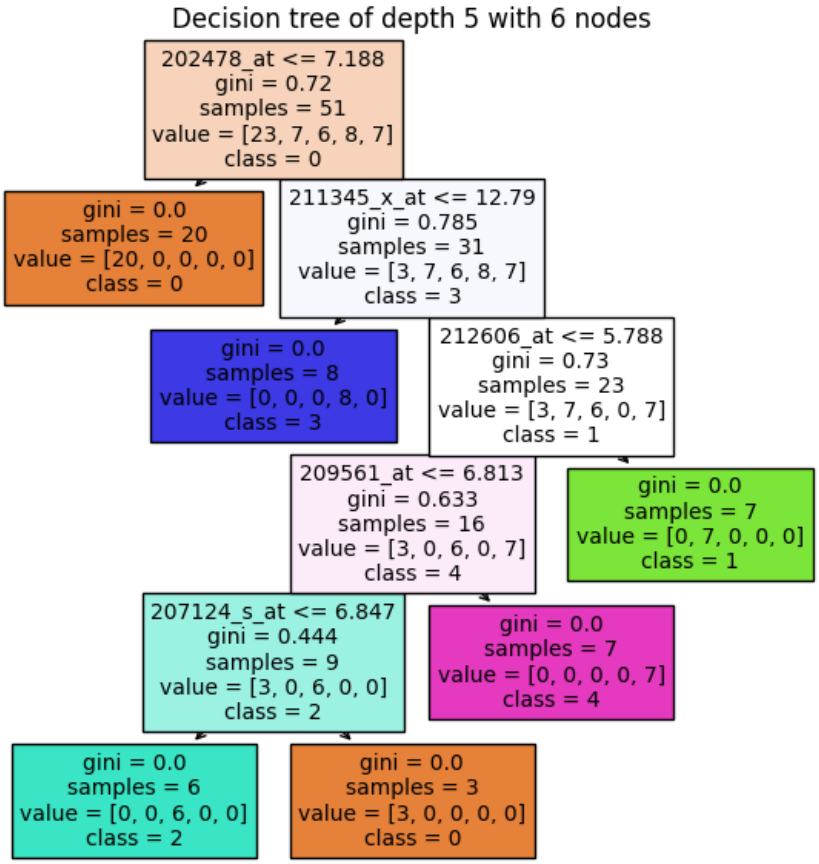
Пре него што кренемо да се бавимо моделима, остала нам је још једна ствар. Наш скуп података ћемо делити у тренинг и тест скупове. За ово ћемо користити train_test_split функцију из библиотеке sklearn. Користићемо 80% података за тренирање и 20% за тестирање модела.
Такође, за неке алгоритме потребно нам је да прво нормализујемо наше податке. То ћемо радити функцијом MinMaxScaler такође из sklearn библиотеке.

Сада смо спремни да се бавимо алгоритмима.

Класификација

Стабла одлучивања

Први алгоритам којим ћемо се бавити јесу стабла одлучивања. Ово је алгоритам надгледаног учења и за њега морамо прво поделити скуп на део за тренинг и део за тест. Користећи функцију DecisionTreeClassifier ћемо направити наше наивно стабло, без икаквог подешавања хипер параметара.

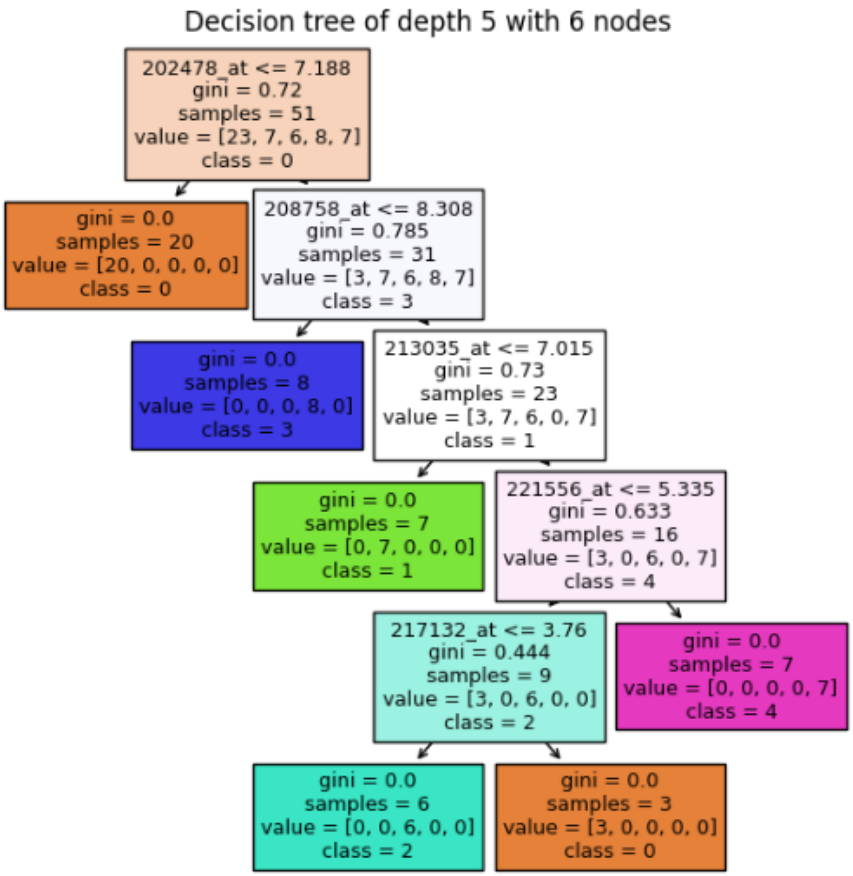


Слика 6: Наивно стабло одлучивања

Примећујемо да је наше стабло изузетно плитко, тј. дубина му је само 5. Али узмимо у обзир да наш скуп садржи експресију 22283 гена и да не могу сви бити битни за одређивање типа леукемије. Напротив, логичније је да постоји јако мали подскуп гена који утичу на развој леукемије и на њен тип.

Друга ствар коју примећујемо је да приликом сваког покретања алгоритма, добијамо другачије стабло.
Овај алгоритам бира атрибуте по којима ће одрадити раздвајање инстанци на два дела, и то ради изнова и изнова док не дође до тренутка кад са одређеном сигурношћу може да тврди да су све инстанце у тренутном листу из исте класе, или док не дође до унапред одређене максималне дубине. Пошто нисмо подешавали никакве хиперпараметре, једино је могуће да наше стабло налази већи број подједнако добрих атрибута по којима ће да врши поделе и приликом сваког покретања, насумице бира један од њих. Наравно, сваким покретањем алгоритма, уз ново стабло добијамо и нове перформансе.

Ово можемо променити тако што додамо параметар random_state нашем алгоритму. Овај параметар ће загарантовати да при сваком покретању алгоритма, добијамо исто стабло са истим перформансама.
Сазнајемо да највеће перформансе наш алгоритам има за random_state = 2, па ћемо сада конструисати стабло само са тим параметром.



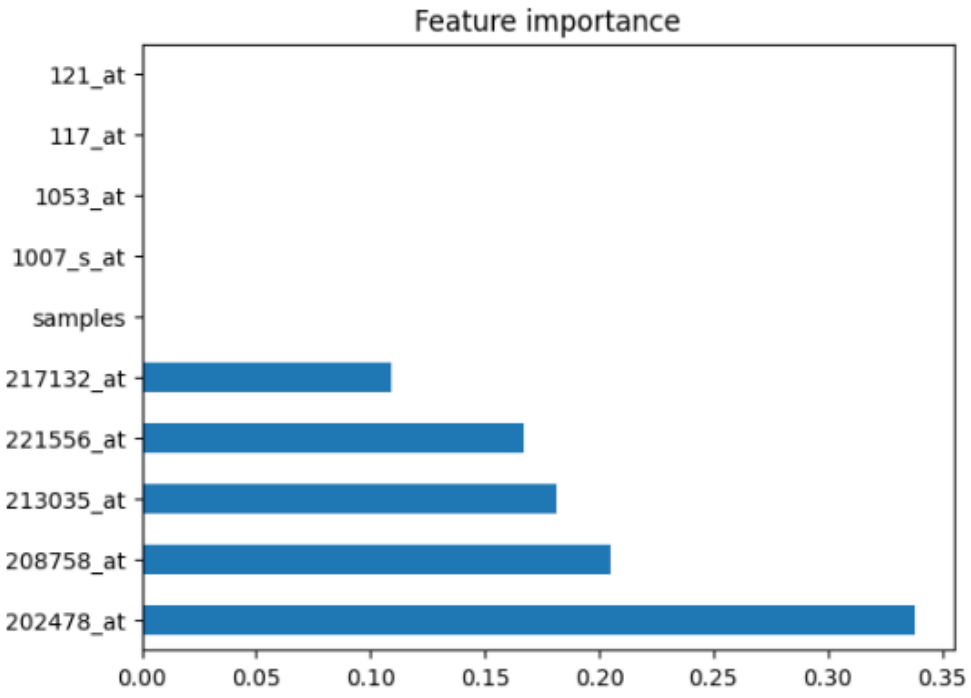
Слика 7: Наивно стабло са најбољим перформансама

Ово стабло има прецизност око 85%.

```
------------------------------------------------------------------------
Classification report for model DecisionTreeClassifier on test data
------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.67      0.67      0.67         3
           1       1.00      1.00      1.00         3
           2       0.67      1.00      0.80         2
           3       1.00      1.00      1.00         2
           4       1.00      0.67      0.80         3

    accuracy                           0.85        13
   macro avg       0.87      0.87      0.85        13
weighted avg       0.87      0.85      0.85        13

------------------------------------------------------------------------
Confusion matrix for model DecisionTreeClassifier on test data
------------------------------------------------------------------------
   0  1  2  3  4
0  2  0  1  0  0
1  0  3  0  0  0
2  0  0  2  0  0
3  0  0  0  2  0
4  1  0  0  0  2
------------------------------------------------------------------------
```

Слика 8: Прецизност наивног стабла са најбољим перформансама

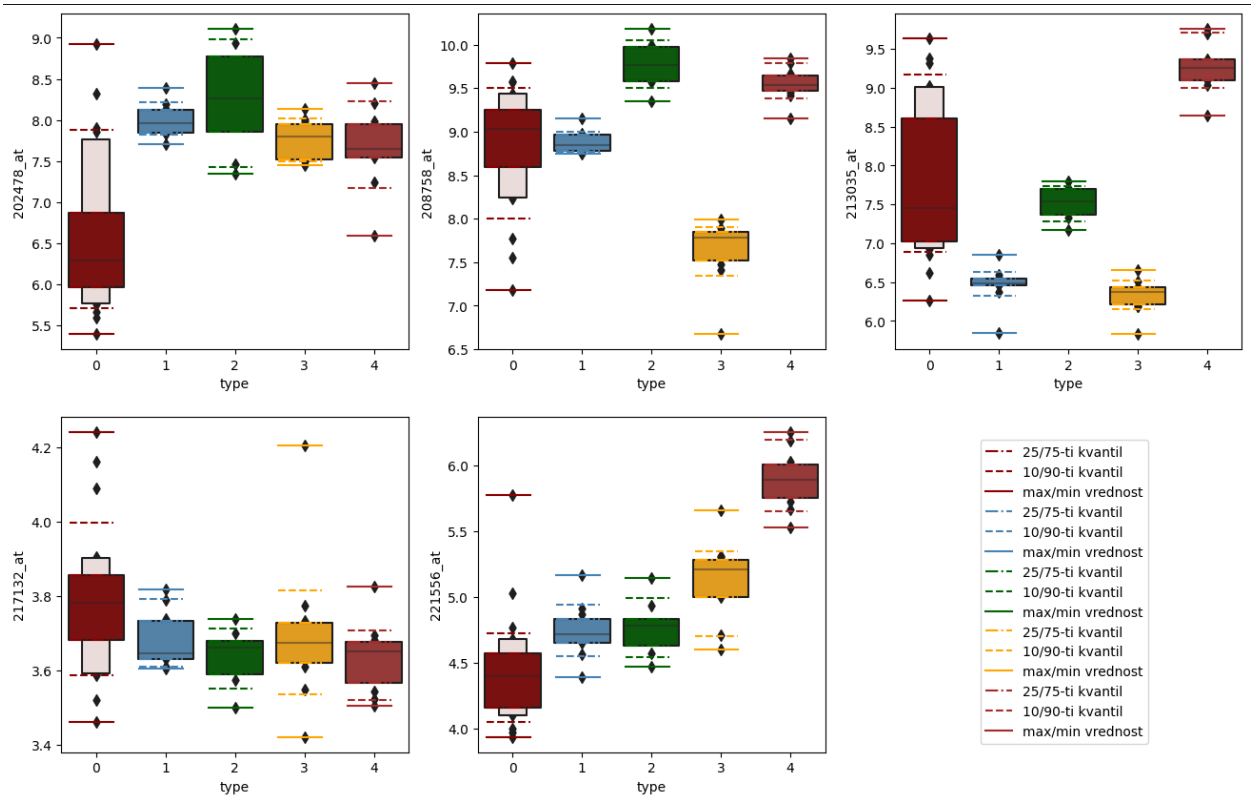
Из овог стабла можемо да извучемо информације о томе којим атрибутима је ово стабло давало колики значај, и то се види на следећој слици.



Слика 9: Значај атрибута

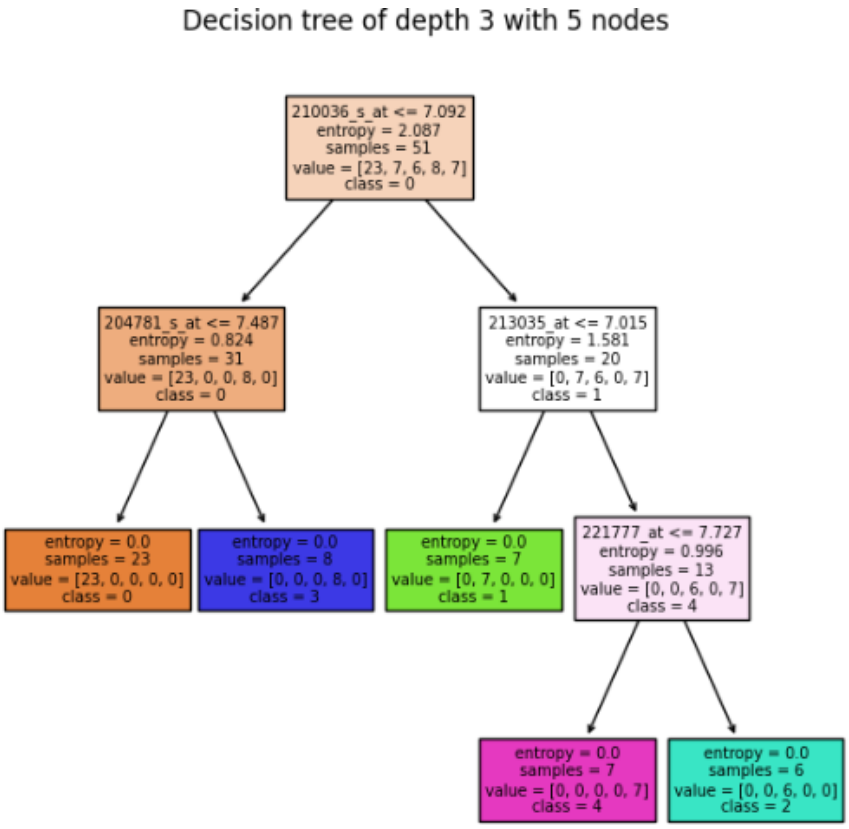
Са овог графика можемо видети да наше стабло придаје значај само оним атрибутима по којима је вршило раздвајање, што је било и очекивано. Највећи значај придаје атрибуту 202478_at.

Можемо и одрадити бокс плот ових атрибута како би визуелно претставили њихову дистрибуцију.



Слика 10: Дистибуција експресија гена који су од највећег значаја

Ако пробамо да нађемо најбоље хиперпараметре помоћу функције GridSearchCV добијемо да су оптимални параметри да максимална дубина буде 4 и да се као критеријум користи ентропија, али ово стабло на тест скупу има горе перформансе од првог.



Слика 11: Стабло са подешеним хиперпараметрима

Classification report for model DecisionTreeClassifier on test data

	precision	recall	f1-score	support
0	0.67	0.67	0.67	3
1	1.00	1.00	1.00	3
2	0.00	0.00	0.00	2
3	1.00	1.00	1.00	2
4	0.25	0.33	0.29	3
accuracy			0.62	13
macro avg	0.58	0.60	0.59	13
weighted avg	0.60	0.62	0.60	13

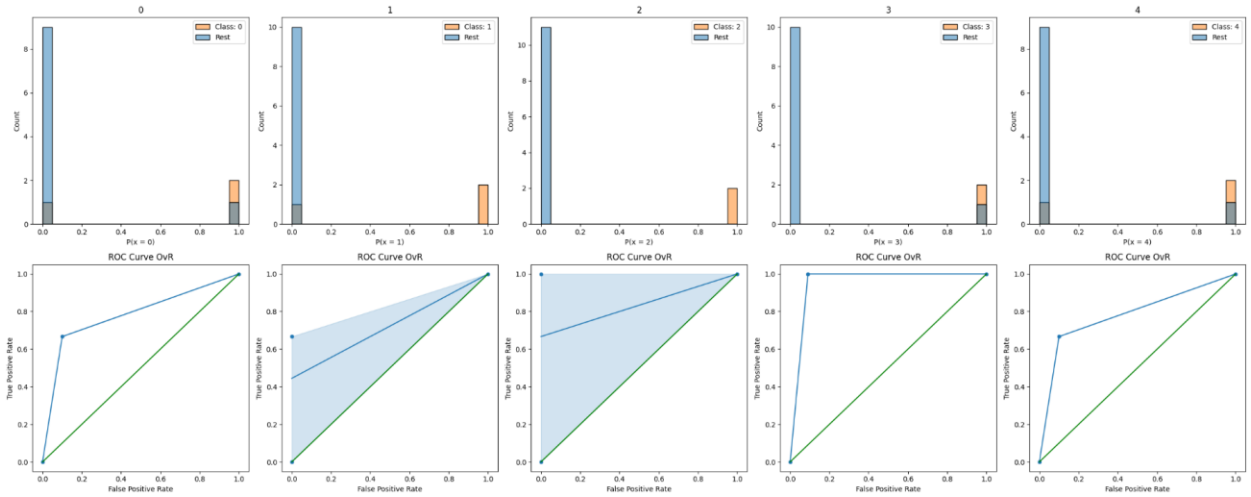
Confusion matrix for model DecisionTreeClassifier on test data

	0	1	2	3	4
0	2	0	0	0	1
1	0	3	0	0	0
2	0	0	0	0	2
3	0	0	0	2	0
4	1	0	1	0	1

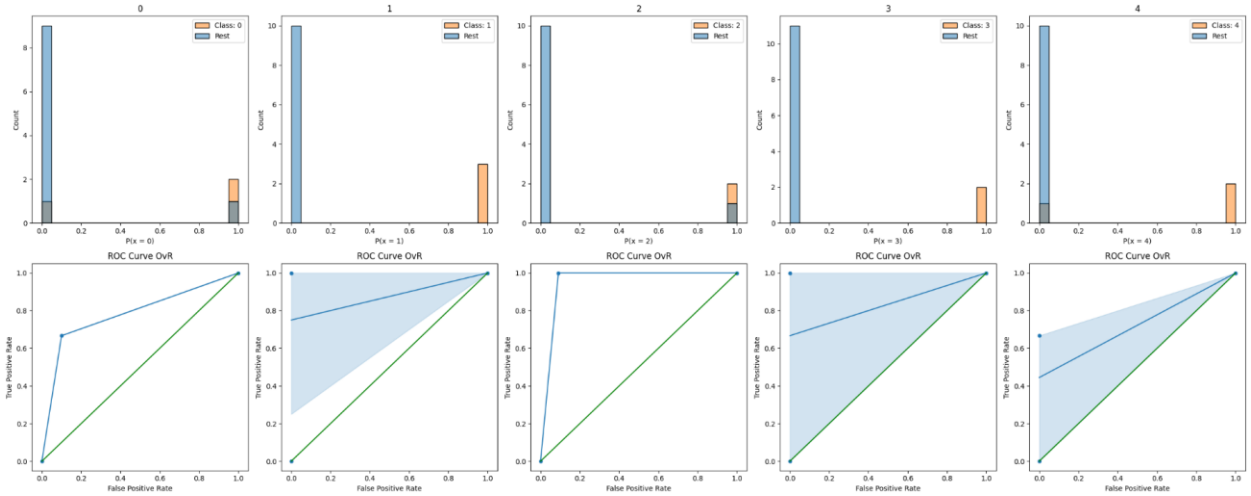
Слика 12: Прецизност стабла

Поређење ових модела можемо извршити помоћу РОЦ криве, али са обзиром да је тај метод намењен да мери ,,раздвојеност“ бинарних класа, морађемо да то радимо методом један против осталих.

Ова метода мери ,,раздвојеност“ једне класе од свих осталих. Ово значи да ћемо морати да меримо пет пута.



Слика 13: Хистограм и роц криве првог наивног модела



Слика 14: Хистограм и роц криве најбољег модела

Види се да је први модел бољи за неке класе (нпр. 3), али је други модел бољи у за неке друге класе (нпр. 1, 2)

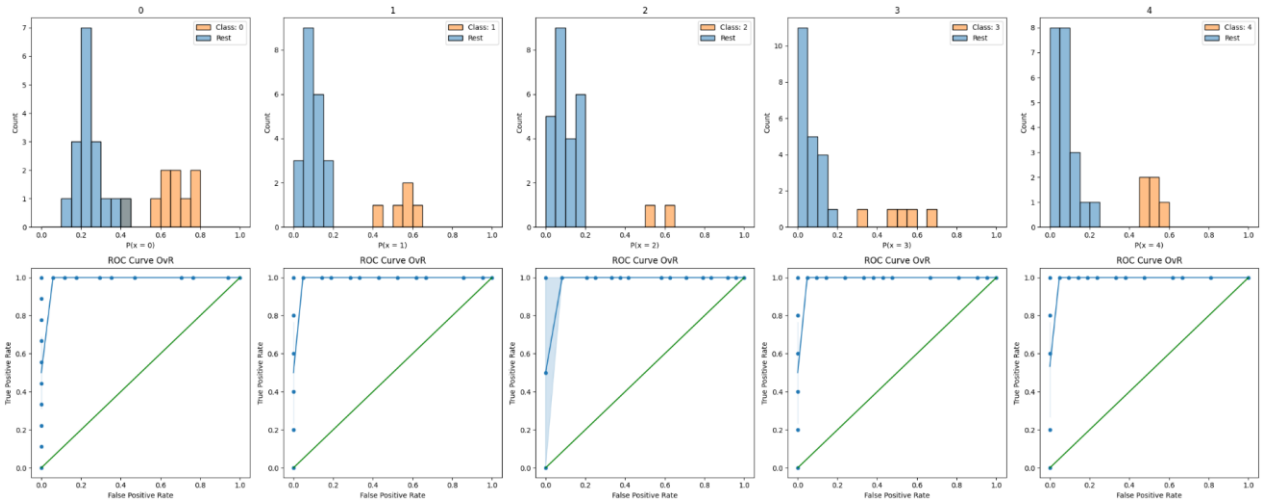
Случајне шуме

Пробаћемо и алгоритам случајних шума, али овај пут ћемо одмах скочити на део тражења најбољих хиперпараметара.
Са оптималним параметрима овај модел на тест скупу има прецизност 96%.

```
--------------------------------------------------------------------------------
Classification report for model RandomForestClassifier on test data
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.90      1.00      0.95         9
           1       1.00      1.00      1.00         5
           2       1.00      1.00      1.00         2
           3       1.00      0.80      0.89         5
           4       1.00      1.00      1.00         5

    accuracy                           0.96        26
   macro avg       0.98      0.96      0.97        26
weighted avg       0.97      0.96      0.96        26

--------------------------------------------------------------------------------
Confusion matrix for model RandomForestClassifier on test data
--------------------------------------------------------------------------------
   0  1  2  3  4
0  9  0  0  0  0
1  0  5  0  0  0
2  0  0  2  0  0
3  1  0  0  4  0
4  0  0  0  0  5
--------------------------------------------------------------------------------
```

Слика 15: Прецизност случајних шума

Сада можемо помоћу роц криве упоредити претходно најбољи модел са случајним шумама.



Слика 16: Хистограм и роц криве случајних шума

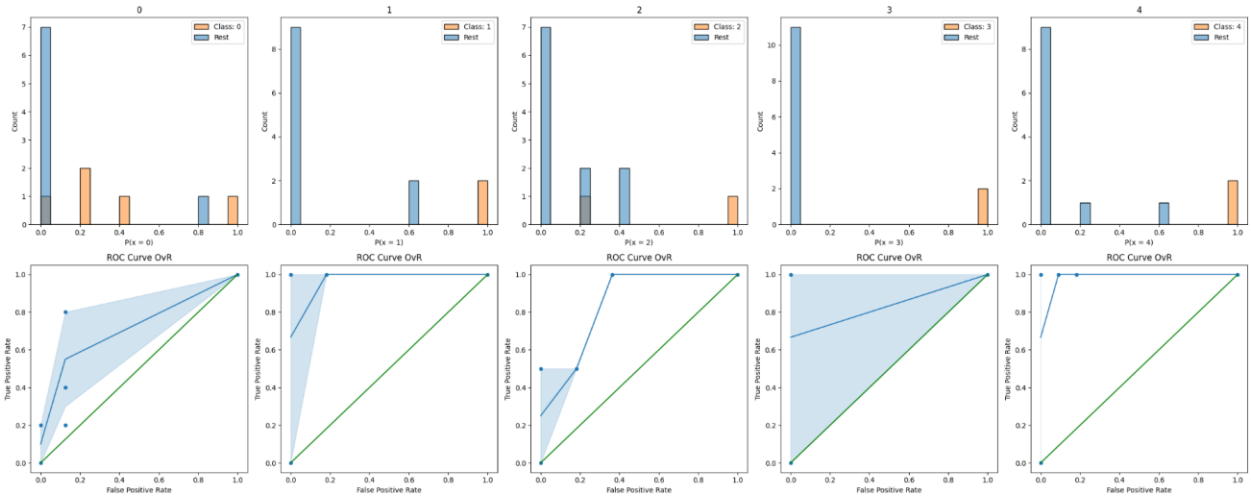
Поређењем роц кривих са слика 14 и 16 видимо да случајне шуме имају највећу ,,раздвојеност“ сваке класе од осталих.

К најближих суседа

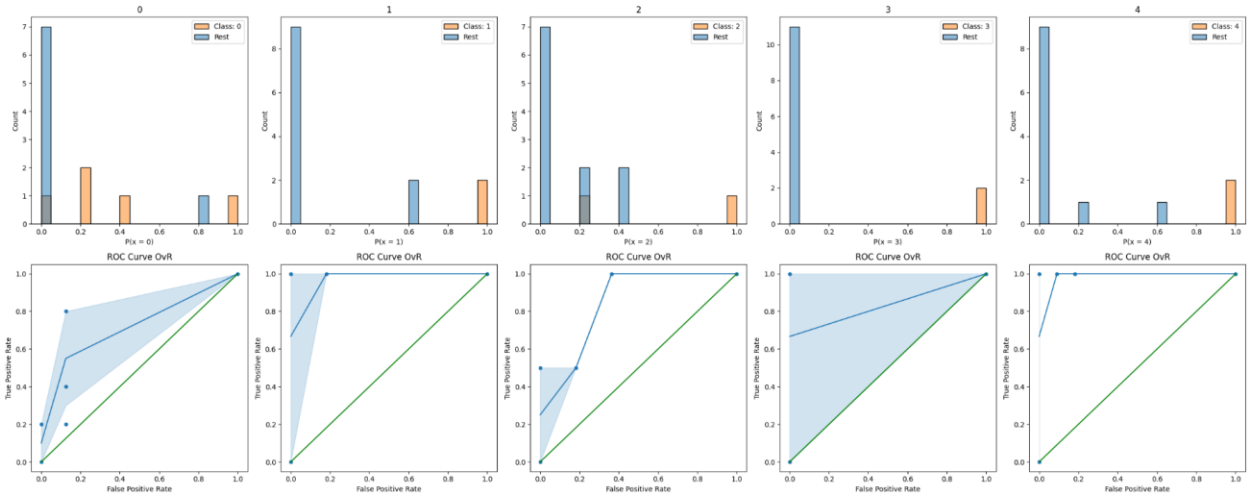
За овај алгоритам је потребно прво нормализовати податке. Овде би могло доћи до проблема због великог броја аутлајера који би покварили нормализацију, али као што је већ речено, ови аутлајери нису превише изван граница, нити су настали због неких грешака приликом уноса података. Свакако, можемо пробати мало специфичан поступак па упоредити два различита модела.

Пошто наш скуп података има много аутлајера, а притом садржи мало инстанци са великим бројем атрибута, ако би избацивали инстанце које имају велики број аутлајера, од нашег скупа података не би ништа остало. Па ћемо пробати да сваки атрибут који има преко 5% аутлајера (њих 3849) избацимо из табеле и тако оставимо све инстанце али са мањим бројем атрибута у односу на почетну табелу. Покренућемо алгоритам на почетним и на модификованим подацима и упоредићемо резултате. Очекивање би било да модел са промењеним подацима буде или исти, или гори од полазног, јер ти додатни подаци могу само бити битни, у ком случају модел постаје гори ако их избацимо, или небитни, у ком случају не праве разлику.

Дакле, сада имамо почетни скуп података и измењени скуп података. Нормализацијом и покретањем КНН алгоритма на оба скупа добијамо идентичне моделе са идентичним прецизностима (69%) и роц кривама.



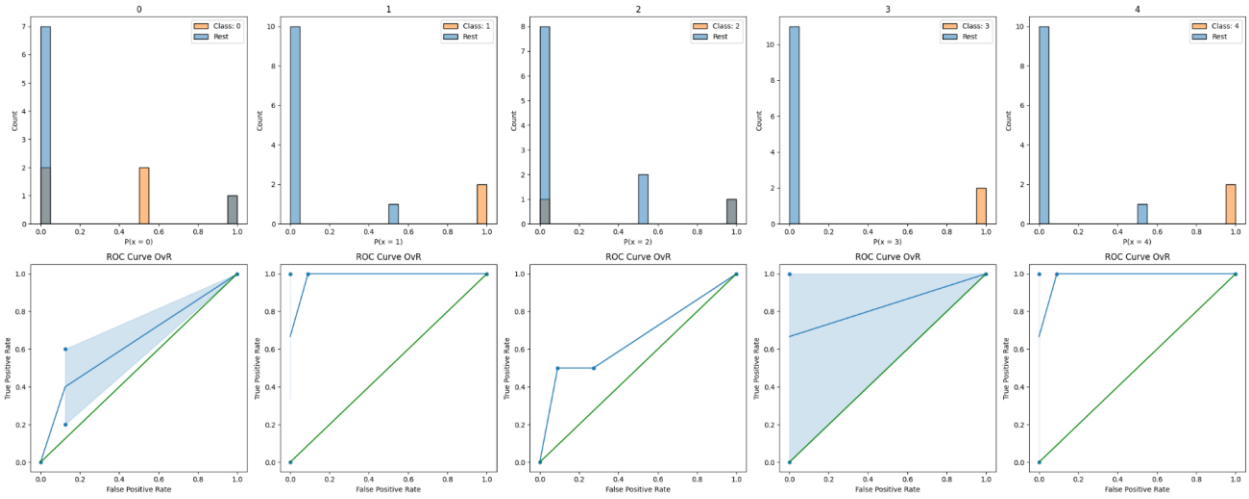
Слика 17: Хистограм и роц криве кнн алгоритма са почетним подацима



Слика 18: Хистограм и роц криве кнн алгоритма са промењеним подацима

Из овога можемо закључити да су модели исти са и без података са много аутлајера, тј. да ти подаци нису битни за одређивање типа леукемије.

Подешавањем хиперпараметара на првом моделу успевамо да добијемо модел који има нешто већу прецизност (77%).



Слика 19: Хистограм и роц криве кнн алгоритма са подешеним хипер параметрма

Од свих ових алгоритама, по роц кривама, најбоље перформансе наизглед има алгоритам случајних шума. Код њега се постиже највећа ,,раздвојеност“.

Кластеровање

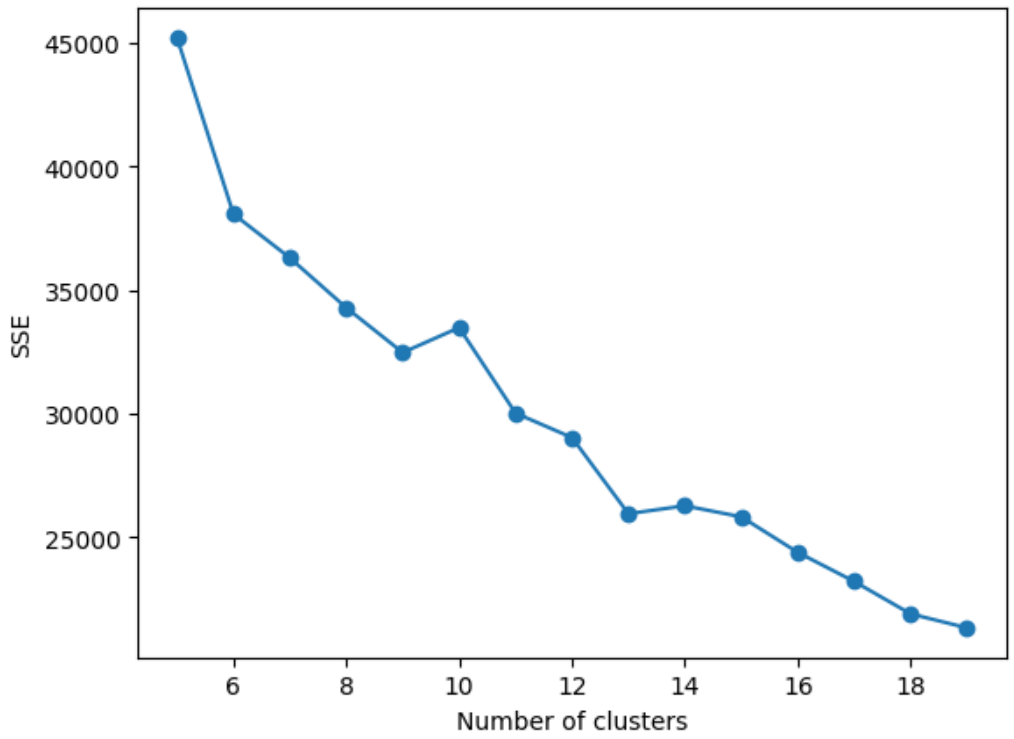
Модели кластеровања су модели ненадгледаног учења. Тј. за њих нећемо делити податке на тренинг и тест скупове и нећемо моћи да проверавамо прецизност.

Алгоритам К средина

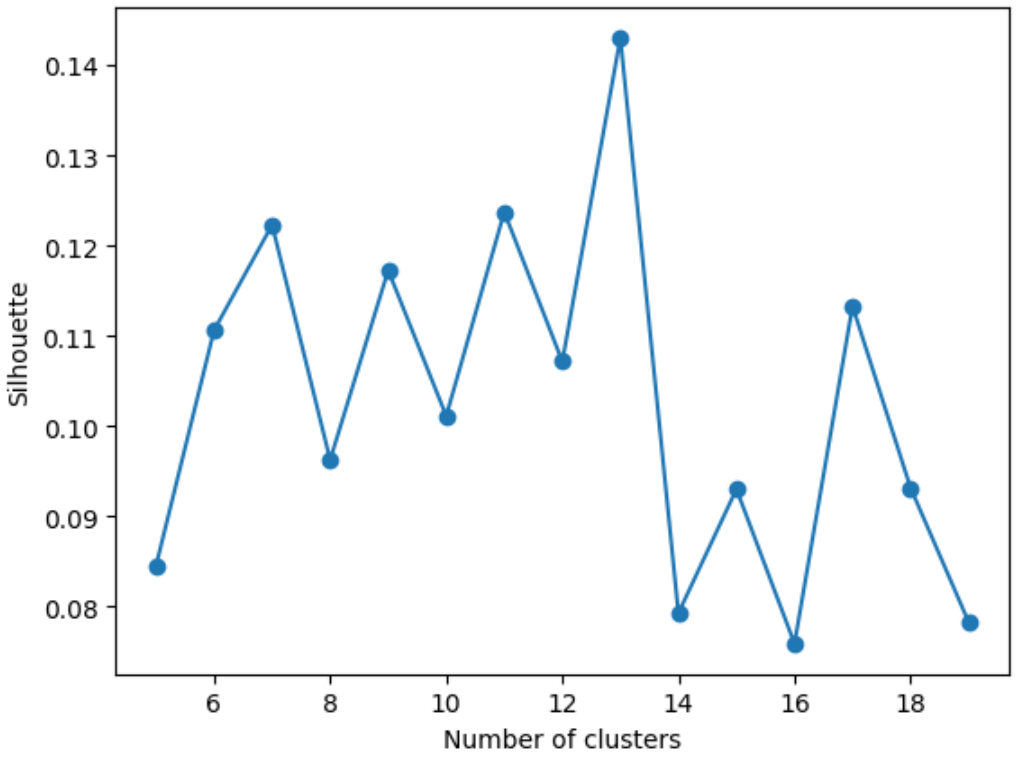
Алгоритам К средина прво насумично бира К центара кластера, а затим у итерацијама придружује сваки податак најближем центроиду и ажурира његове позиције како би се минимизовала удаљеност унутар кластера. Овај процес се врши изнова док се центроиди не стабилизују.

Покретањем овог алгоритма више пута са различитим параметром за број кластера и мерењем силуете и скг (сума квадратних грешака) вредности је један од начина за одабир броја кластера.

Помоћу ПЦА алгоритма смањићемо димензионалност наших података на две димензије како би могли да визуелизујемо промене.



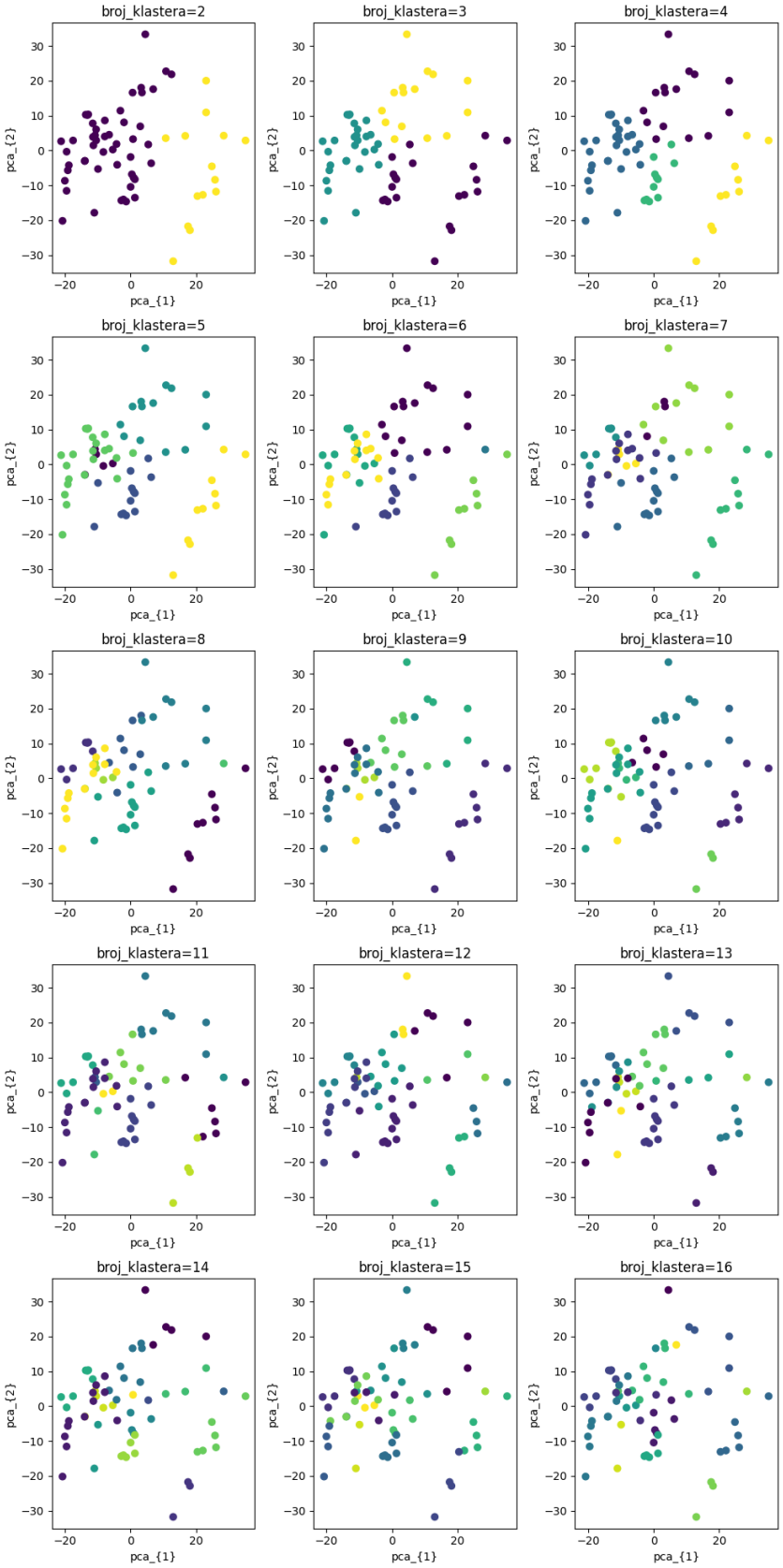
Слика 20: СКГ по броју кластера



Слика 21: Силуета по броју кластера

Када бирамо број кластера, то радимо методом лакта. Тражимо место где СКГ најбрже опада али упарићемо то са са вредности силуете тако да ћемо истовремено желети да нам силуета буде што већа.
Комбинацијом ова два графика видимо да би најбоље било за број кластера изабрати 9 или 13.

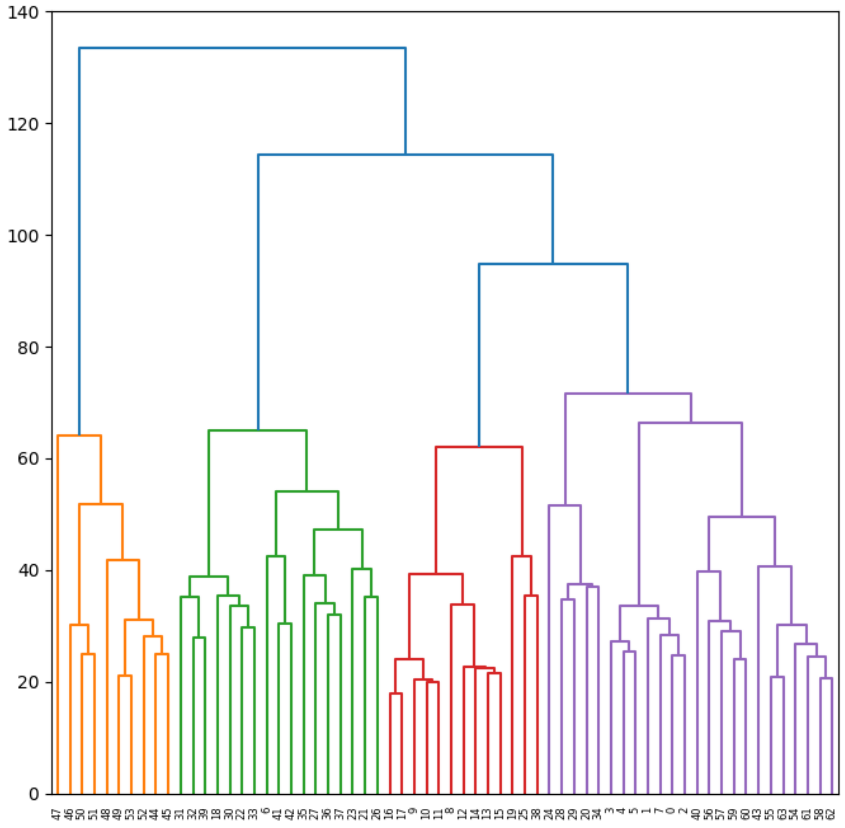
Помоћу ПЦА алгоритма можемо да визуелизујемо како се кластери мењају са повећањем њиховог броја.



Слика 22: Промене са повећањем броја кластера

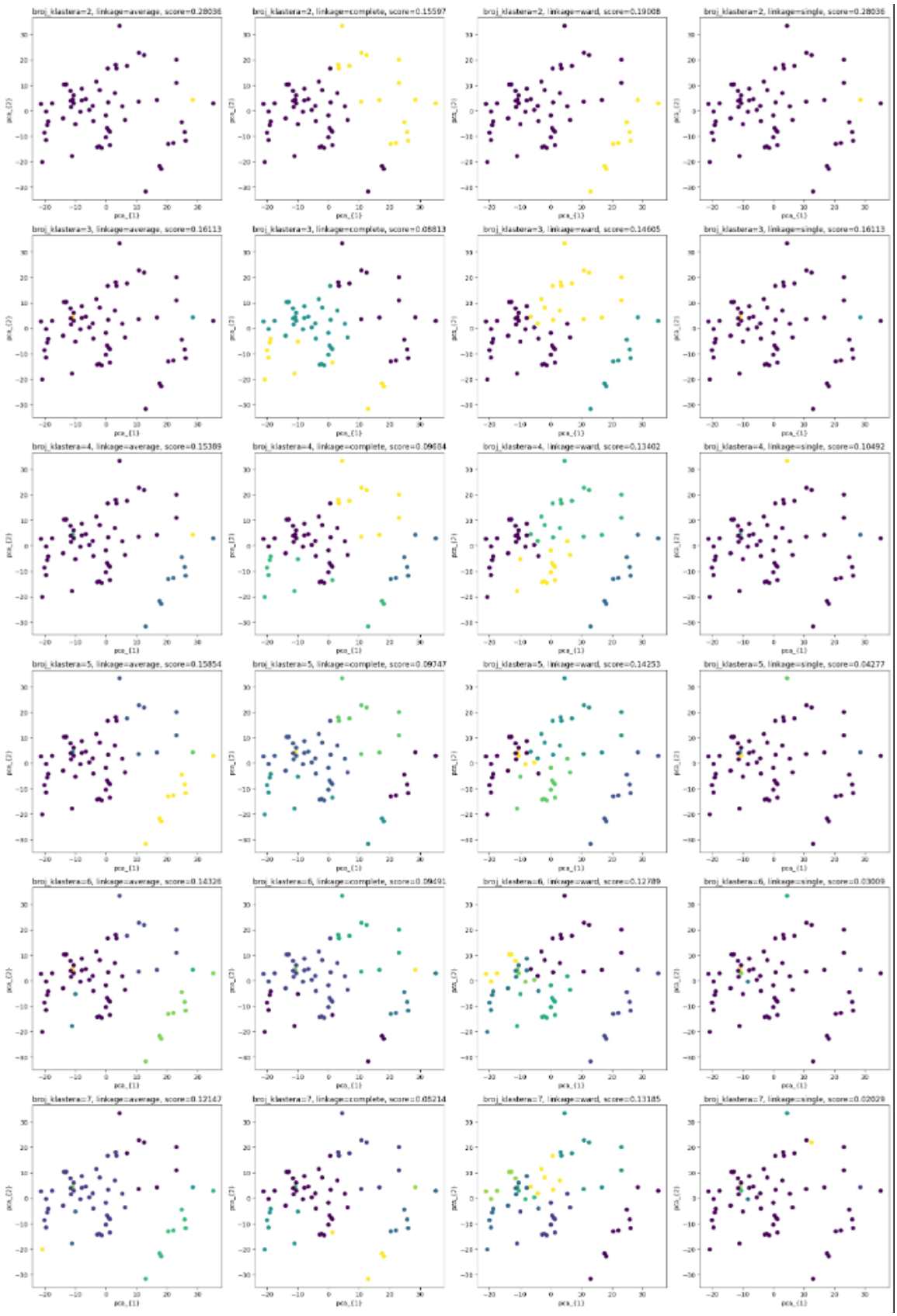
Хијерархијско кластеровање

Још један начин кластеровања је хијерархијско сакупљајуће кластеровање. Овај алгоритам проналази два најближа елемента, групише их у један кластер и тај процес врти у круг са тиме да се претходно груписане инстанце сматрају једним елементом. Раздаљина између објеката се може рачунати на више начина, овде је коришћен Вардов метод. На наредној слици видимо добијену хијерархију.

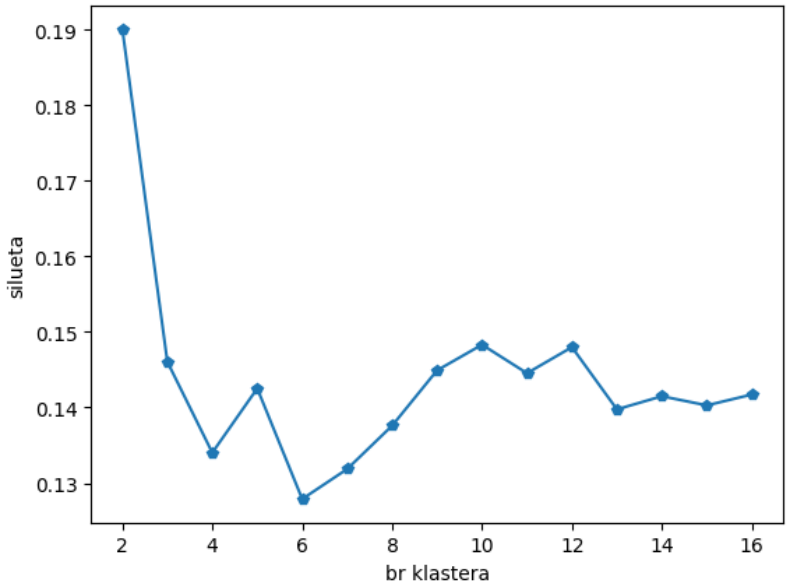


Слика 23: Визуелни приказ хијерархијског кластеровања

На следећој слици видимо плотове који се разликују по критеријуму и по броју кластера. Можемо видети да најбољи скор силуете имамо за 2 кластера и критеријум просека.

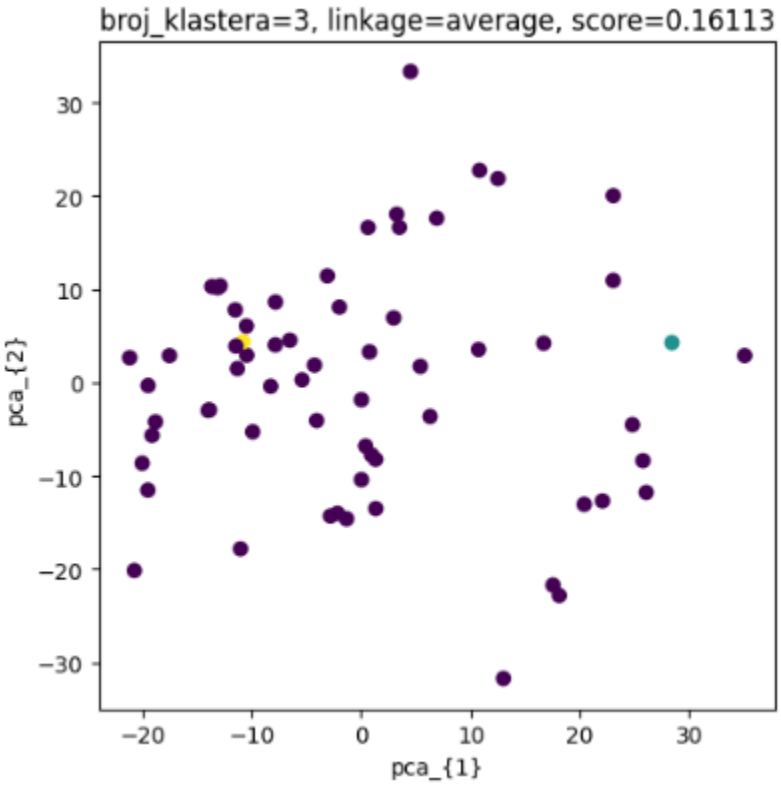


Слика 24: Промене са повећањем броја кластера и различитим критеријумима



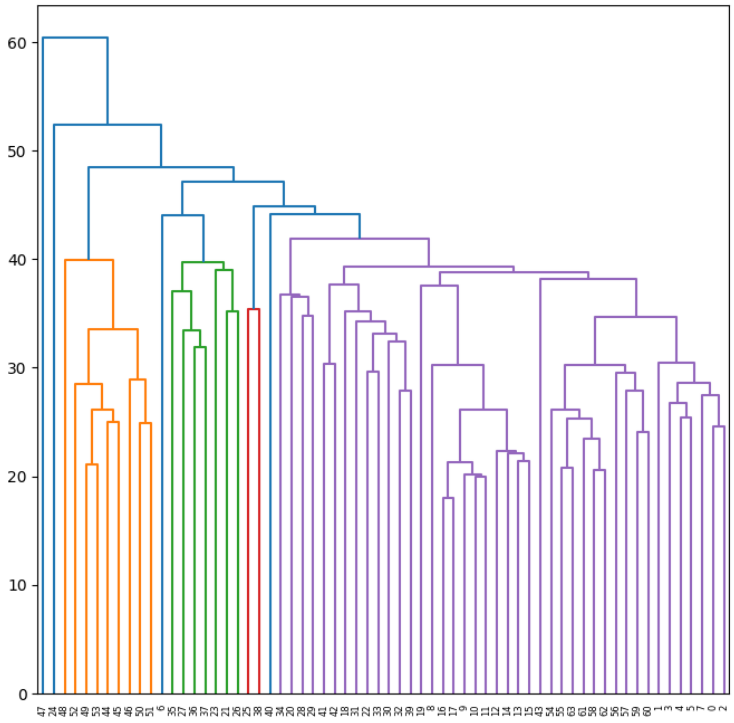
Слика 25: Промена силуете по броју кластера

Силуета има највећу вредност када имамо 2 кластера, али пошто желимо већи број од тога, изабраћемо следећи најбољи, тј. 3.



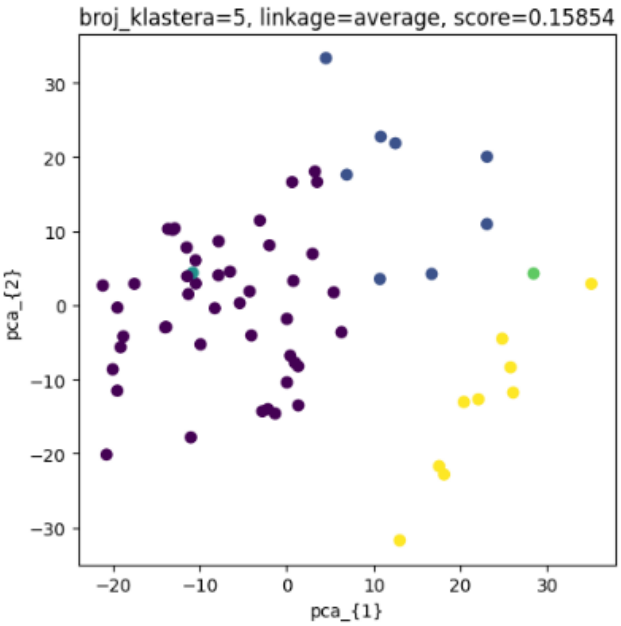
Слика 26: Три кластера добијени хијерархијским кластеровањем

Дендрограм ове хијерархије можемо видети на следећој слици.



Слика 27: Визуелни приказ хијерархије

Следећи најбољи скор би био за критеријум просека и 5 кластера.



Слика 28: Пет кластера добијени хијерархијским кластеровањем

Правила придруживања

Користећи априори принцип генерисања правила придруживања покушаћемо да нађемо нека правила.

Прво, морамо дискретизовати наше атрибуте. Дискретизоваћемо податке у 7 корпи, на +/-3 стандардне девијације. Атрибути који се налазе у -3 интервалу ће као своју дискретну вредност добити баш -3, и тако све до +3.

Сада када имамо дискретизоване податке, потребно је балансирати класе јер у нашем скупу има више примерака АМЛ леукемије па би онда и правила придруживања за тај тип била бројнија и имала би већу подршку. Балансирање радимо оверсемпловањем.

На оваквим подацима априори алгоритам нам налази велики број правила. Нека од правила се могу видети на следећој слици.

Consequent	Antecedent	Support %	Confidence %	Lift
type = Bone_Marr...	200043_at_SDBI... 200005_at_SDBI...	20.0	92.308	4.444
type = Bone_Marr...	200043_at_SDBI... 200013_at_SDBI...	20.0	92.308	4.444
type = Bone_Marr...	200043_at_SDBI... 200022_at_SDBI...	20.0	92.308	4.444
type = Bone_Marr...	200043_at_SDBI... 200031_s_at_SD...	20.0	92.308	4.444
type = Bone_Marr...	200043_at_SDBI... 200018_at_SDBI...	20.0	92.308	4.444
type = Bone_Marr...	200043_at_SDBI... 200005_at_SDBI... 200013_at_SDBI...	20.0	92.308	4.444
type = Bone_Marr...	200043_at_SDBI... 200005_at_SDBI... 200022_at_SDBI...	20.0	92.308	4.444
type = Bone_Marr...	200043_at_SDBI... 200005_at_SDBI... 200031_s_at_SD...	20.0	92.308	4.444
type = Bone_Marr...	200043_at_SDBI... 200005_at_SDBI... 200010_at_SDBI...	20.0	92.308	4.444
type = Bone_Marr...	200043_at_SDBI... 200005_at_SDBI... 200024_at_SDBI...	20.0	92.308	4.444

Слика 29: Правила придруживања

Закључак

Из добијених резултата и модела можемо видети да предвиђање само типа леукемије по експресијама гена уопште није наиван посао. Грешке модела у оваквим случајевима би биле јако скупе, потпуно неприхватљиве, а са обзиром да ни један од ових модела није имао 100% прецизност у одређивању типа леукемије, ни један не би могао бити коришћен у озбиљне сврхе.

Дати скуп података је јако обиман. Приказивање података у облику који би био јасан човеку је у био изазов. Сваки покушај таквог приказа би се завршио губљењем смислености података.

Референце