УНИВЕРЗИТЕТ У БЕОГРАДУ

МАТЕМАТИЧКИ ФАКУЛТЕТ

Милица Чугуровић

УВОД У МАШИНСКО УЧЕЊЕ КОРИШЋЕЊЕМ ПРОГРАМСКОГ ПАКЕТА ORANGE

МАСТЕР РАД

Београд, 2021.

Ментор:

др Мирослав Марић, ванредни професор

Универзитет у Београду, Математички факултет

Чланови комисије:

др Младен Николић, доцент

Универзитет у Београду, Математички факултет

др Александар Картељ, доцент

Универзитет у Београду, Математички факултет

Датум одбране: 17. 9. 2021.

Наслов мастер рада: Увод у машинско учење коришћењем програмског пакета Orange

Резиме: Вештачка интелигенција и машинско учење као њена подобласт представљају једну од тренутно најпопуларнијих области рачунарства. Апликације засноване на машинском учењу своју примену налазе у решавању великог броја проблема, при чему је њихов утицај на интеракцију људи са машинама све већи. Имуајући у виду растући значај машинског учења, било би добро што већем броју корисника представити његове основне концепте.

Циљ овог рада је упознавање са основним појмовима машинског учења. За потребе рада биће креирани материјали који не захтевају познавање напреднијих програмских језика. Материјали ће бити погодни за кориснике који се тек упознају са уводним концептима програмирања, као и за употребу у основним и средњим школама. Основни појмови машинског учења ће бити уведени коришћењем програмског окружења Orange, које је засновано на превуци и пусти парадигми (енг. *drag and drop*). У раду ће бити обрађене теме које се односе на податке, њихов значај у машинском учењу, припрему и визуализацију. Након тога ће бити представљени модели машинског учења уз помоћ којих се врши анализа података, начини креирања ових модела, њихова употреба и евалуација.

Кључне речи: вештачка интелигенција, машинско учење, класификација, регресија, неуронске мреже, кластеровање

Садржај

Глава 1: Увод - основни појмови машинског учења

1.1 Вештачка интелигенција

Вештачка интелигенција представља подобласт рачунарства која подразумева развој софтвера који ће рачунаре направити способним за интелигентно понашање: усвајање, памћење и обраду одређених знања. Симулирање интелигенције коришћењем машина, насупрот природној интелигенцији која је карактеристична за жива бића – људе и животиње, пред собом има бројна ограничења. Највеће од њих јесте то како превазићи јаз између биолошког и вештачког. Термин вештачка интелигенција неретко се употребљава за описивање машина које интерпретирају функције својствене човеку – учење, како индуктивно (од посебног ка општем) тако и дедуктивно (од општег ка посебном). Међутим, иако би многи помислили да је основни циљ вештачке интелигенције опонашање људског начина учења, већина подобласти вештачке интелигенције има сасвим другачији циљ: решавање проблема у којима је доминантан проблем комбинаторна експлозија простора решења, учење на основу скупова података велике димензије, обрада скупова података велике димензије и слично.

Област вештачке интелигенције заснована је на претпоставци да је људска интелигенција таква да се може на довољно добар начин симулирати и да се на основу тога може конструисати машина која ће је симулирати. Један конкретан и често употребљаван пример наведеног су неуронске мреже, које се заснивају на опонашању функционисања неурона у људском мозгу. Вештачка интелигенција се заснива на рачунарској науци, информационим технологијама, математици, статистици, психологији, лингвистици и многим другим областима.

Вештачка интелигенција је технологија која већ утиче на интеракцију како људи међусобно, тако и људи са машинама. Иако већ сада ова област има велики утицај на све сфере живота, у блиској будућности њен ће утицај извесно наставити да расте. Ова област има потенцијал да увелико измени начин на који људи комуницирају, не само са дигиталним светом, већ и међусобно.

Апликације засноване на вештачкој интелигенцији већ се примењују у здравственој дијагностици, лечењу, превозу, јавној безбедности, образовању и забави, али ће се применити на више области у наредним годинама. Такође, област вештачке интелигенције налази примену у областима у којима је то било најмање очекивано, и остварује резултате који увелико надмашују досадашња достигнућа и предвиђања. Заједно са интернетом, вештачка интелигенција мења начин на који доживљавамо свет и има потенцијал да буде нови покретач економског раста.

1.2 Машинско учење

Машинско учење је грана вештачке интелигенције заснована на идеји да системи могу да уче из података, идентификују обрасце и доносе одлуке самостално, без експлицитног програмирања од стране човека. Уместо да човек програмира сваки корак, овај приступ даје рачунарима моћ да уче и закључују из података без детаљних упутстава од стране програмера. То значи да се рачунари могу користити за нове, компликоване задатке који се нису могли "ручно" програмирати - ствари попут апликација за препознавање лица на видео материјалима или фотографијама или превођење слика у говор.

Машинско учење погодно је за примену у ситуацијама када се решавају проблеми које је могуће врло лако формулисати и савладати (као што је рецимо препознавање лица на сликама) али са друге стране кораке који доводе до решења није могуће лако експлицитно дефинисати. Овде се ради о томе да је препознавање лица човек усавршио на основу искуства, а искуство је могуће формализовати подацима. Дакле, искуство које је човек стекао гледајући лица око себе свакога дана могуће је представити методама машинског учења великим бројем слика на којима се налазе лица. Тада ће методи машинског учења, слично као и људи, на основу њих покушати да донесу закључке. На основу претходног јасно је да подаци заузимају важно место приликом дизајнирања модела машинског учења.

Податке је најчешће потребно припремити и прилагодити, потребно је пронаћи најбољи начин за достављање података методу. Важно је нагласити да постоје и методи машинског учења који су толико моћни да су способни да уче и из сирових, то јест неприпремљених података, али су свакако чешћи методи који захтевају добро припремљене и прилагођене податке. Натрениран модел машинског учења затим уочава законитости које важе на датом скупу података. Ово је у суштини генерисање новог алгоритма, формално названог моделом машинског учења. Коришћењем различитих података при обуци, подешавањем параметара који чине сам модел, исти метод учења могао би се користити за генерисање различитих модела.

Иако метод машинског учења може примењивати комбинацију различитих техника, методе учења се обично могу категорисати у три општа типа.

1. **Надгледано учење** (енг. *supervised learning*): методу се дају подаци на основу којих учи али и жељени излази за дате податке (оно што је потребно научити). На основу тога метод мора да научи да за дате податке одреди одговарајуће излазе. Пример је класификација цветова, где је скуп података на којима модел учи такав да је уз сваку слику дата и информација о врсти цвета који се на њој налази.

2. **Ненадгледано учење** (енг. *unsupervised learning*): подаци дати методу учења су неозначени, односно нису упарени са одговарајућим излазима (циљним вредностима). Од метода се тражи да идентификује обрасце у датим подацима. Пример је груписање већег броја тачака еуклидске равни у неколико група на основу њиховог међусобног растојања, што пак представља меру њихове сличности.

3. **Учење поткрепљивањем** (енг. *reinforcement learning*): ова метода симулира процес учења користећи принцип награде и казне. Можда напознатија примена ове методе машинског учења јесу самовозећи аутомобили који су базирани на техникама машинског учења где се уз помоћ различитих сензора и камера добијају подаци о околини и на основу тога доносе одлуке о кретању возила у простору. На основу донетих одлука моделу се додељује награда или казна. Модел тежи таквим одлукама које максимизују вредност добијене награде.

1.3 Тренутна достигнућа и примене машинског учења

Иако термин вештачка интелигенција многе асоцира на научну фантастику, ова област је већ дуго заступљена у научним круговима, док се у последње време интензивно развија највише захваљујући њеној подобласти – машинском учењу. Ова област већ има широке примене, а њихови примери наведени су у наставку.

1. Филтрирање е-поште: услуге е-поште користе машинско учење за филтрирање долазних порука. Метод сам препознаје одређене поруке као отпад или спам, а тако да корисници немају додир са нежељеним порукама.
2. Персонализација: многе интернет услуге користе вештачку интелигенцију како би персонализовале корисничко искуство. Услуге попут Амазона или Нетфликса уче на основу претходних куповина корисника, као и куповина других корисника (који су по неком критеријуму слични нама) како би препоручиле релевантан садржај.
3. Откривање превара: банке користе алгоритме машинског учења како би утврдиле да ли на корисниковом рачуну постоји неуобичајена активност. Неочекиване активности попут страних трансакција, више трансакција за редом, трансакција са већим износом од уобичајеног могу бити препознате алгоритмом.
4. Препознавање говора: апликације користе машинско учење за оптимизацију функција препознавања говора. Примери укључују интелигентне личне асистенте, на пример Амазонова Алекса или Еплова Сири.
5. Препознавање лица на сликама, препознавање лица на камерама, откључавање телефона детекцијом лица и слично.
6. Машинско превођење текста, препознавање рукописа, превођење са једног језика на други и слично.
7. Аутономна возила. Када је реч о аутономним возилима, потребно је научити модел да, предузимањем одређених радњи (притискања гаса и кочнице, померањем волана и слично) превезе возило од тачке А до тачке Б. Модел треба да опажа стање у ком се тренутно налази, и у складу са тим одлучи коју од поменутих радњи ће предузети.

Претходни примери наводе само мали део онога што машинско учење данас јесте.

1.4 Предности и мане машинског учења

Када је реч о предностима које у живот људи уноси коришћење машинског учења, није тешко закључити да су оне бројне. Предвиђа се да ће утицај машинског учења све више расти, и да ће оно у будућности заузети веома значајно место у области рачунарства. У наставку је наведено пар најбитнијих предности машинског учења.

1. Лако идентификује обрасце: машинским учењем могу се обрадити велике количине података и открити специфични трендови и обрасци који људима не би били видљиви. На пример, за е-трговине као што је Амазон, машинско учење служи за разумевање понашања корисника на мрежи и историје њихове куповине како би им понудили релевантне рекламе.
2. Боље резоновање у односу на човека: модели на основу података могу да генеришу програме какве човек не би могао да напише. Другим речима, способност уочавања правила, повезаности и законитости, као и учења из истих надмашује људске капацитете, нарочито када се ради о великој количини података.
3. Потреба за људском интервенцијом је смањена (аутоматизација): при коришћењу машинског учења није потребно пратити пројекат на сваком кораку. Оно омогућава машинама да предвиђају и такође да у великој мери самостално побољшавају алгоритме. Чест пример за то су антивирусни софтвери који уче да филтрирају нове претње чим су препознате.
4. Непрекидно усавршавање: с обзиром да алгоритми машинског учења стичу искуство, они се уз људску интервенцију побољшавају у тачности и ефикасности са обрадом веће количине података. Ово им омогућава да доносе боље одлуке. Како количина података расте, тако алгоритми боље уче и дају тачније резултате.
5. Широка примена апликација: апликације засноване на техникама машинског учења имају веома широку примену – у здравству, трговини, науци, просвети, аутомобилској индустрији и слично.

Уз све наведене предности у погледу своје снаге и популарности, машинско учење није савршено. У наставку су наведени фактори који ограничавају ову област.

1. Набавка података: већина метода машинског учења, попут дубоких неуронских мрежа, захтева огромне скупове података за тренирање, а они би при томе требали да буду непристрасни и доброг квалитета. Овакви подаци нису лако доступни што представља ограничавајући фактор у овој области.
2. Време и ресурси: потребно је одређено време да се модели натренирају и развију довољно да испуне своју сврху са знатном количином тачности и релевантности. За учење односно тренинг модела су такође потребни огромни хардверски ресурси. Овај проблем се, условљен експанзијом ове области, у великој мери превазилази.
3. Тумачење резултата: још један велики изазов је способност тачне интерпретације резултата генерисаних алгоритмима. Неопходно је и пажљиво бирати алгоритме који ће дати најбоље резултате за одређени проблем а при томе задржати потребну количину интерпретабилности.

4. Велика осетљивост на грешке: с обзиром на то да је машинско учење у великој мери аутономно, оно је веома подложно грешкама. Уколико је скуп за тренирање такав да не осликава на прави начин податке који ће бити коришћени приликом примене модела (обично уколико није довољно велики), резултат су предвиђања пристрасна на скупу за тренирање, која одступају од стварних вредности. У случају машинског учења, такве грешке могу покренути ланац грешака које могу остати неоткривене на дужи временски период; а када буду примећене, потребно је доста времена да се препозна извор проблема, а затим и да се уочени проблем исправи.

Глава 2: Улога и значај скупа података у машинском учењу

2.1 Скуп података

Подаци су, уз алгоритме учења, најважнији део читаве аналитике података па и самог машинског учења. Без података није могуће обучити ниједан модел и сва модерна истраживања области машинског учења и аутоматизације би била узалудна, јер се она базирају управо на подацима. Велике компаније троше значајну количину новца само да би прикупиле и означиле што више података. Додатни напори и ресурси се улажу такође и за чување података, као и за њихово представљање у најпогоднијем облику.

Од почетног одабира модела до завршетка комплетног поступка, користе се три различита скупа података: скуп за тренирање (обуку), скуп за проверу (валидацију) и скуп за тестирање.

1. **Скуп података за тренирање** је део података који се користи за обучавање модела. Ово су подаци које модел користи при обуци (и улаз и излаз у случају надгледаног учења, а у случају ненадгледаног учења само улазни подаци), и на основу којих учи. На основу овог скупа података модел закључује како на основу присутних атрибута донети одлуке (о класификацији, предвиђању, груписању и слично).
2. **Скуп података за валидацију** представља део података који се користи за оцењивање модела још у фази тренирања истог. Користи се да би се избегло преприлагођавање, када је реч о методима машинског учења који се могу подешавати, што углавном и јесте случај. Постоје и напредније технике унакрсне валидације, о којима ће бити речи касније.
3. Након што је модел потпуно обучен, **скуп података за тестирање** пружа непристрасну процену. Важно је да скуп података за тестирање буде потпуно изолован, односно да се не користи у корацима тренирања и валидације. Након уношења улазних података за тестирање, модел ће предвидети неке вредности (без информације о стварном излазу). Након предвиђања, модел се оцењује тако што се упоређује са стварним излазом који је присутан у подацима за тестирање. Дакле, овај скуп података користи се за процену колико је метод обучен на скупу података о тренингу добар, и за оцену колико ће добро модел да се понаша у пракси.

Подаци су веома битан сегмент у процесу машинског учења. Само прикупљање података није довољно. У свим пројектима машинског учења највише пажње треба посветити класификовању и обележавању, односно само класификовању скупова података у случају ненадгледаног учења, јер је неопходно да они буду довољно тачни да одражавају реалну слику онога шта се моделује. Стога ће овоме бити посвећена посебна пажња у наредним главама.

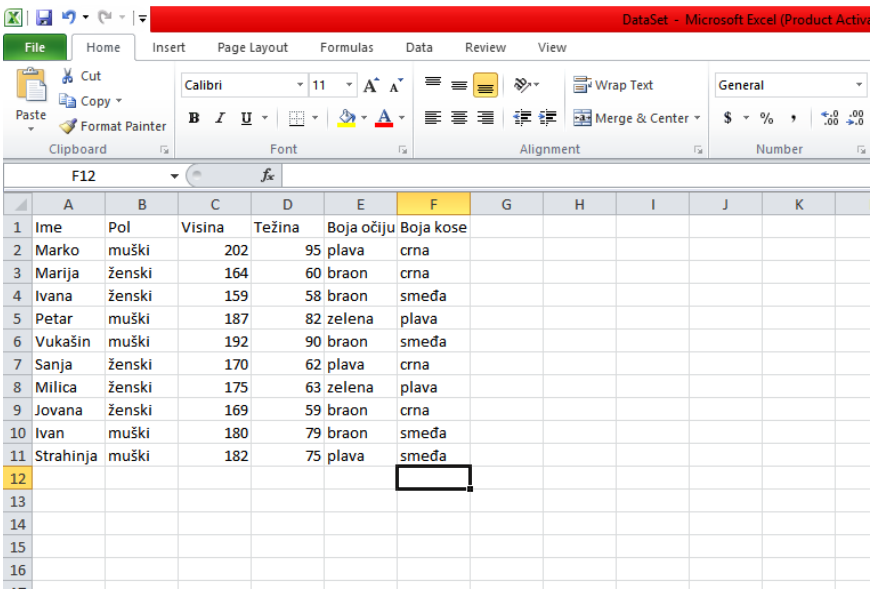
За илустровање основних појмова и концепата машинског учења у овом раду биће коришћено програмско окружење *Orange* [4]. *Orange* је јавно доступан алат за визуелизацију података, анализу података и машинско учење, развијен од стране Универзитета у Љубљани. Компоненете које се користе у програмском окружењу *Orange* могу се користити за визуелизацију података, избор података и предпроцесирање, евалуацију модела и слично. Визуелно програмирање је имплементирано кроз интерфејс у коме се програми креирају повезивањем унапред дефинисаних

компоненти. Подразумевана инсталација укључује бројне алгоритме машинског учења, предпроцесирања и визуализације података у 6 скупова виџета - *data, visualize, classify, regression, evaluate* и *unsupervised.*

2.2 Креирање и учитавање скупова података у окружење *Orange*

Orange окружење подржава различите формате података: Excel документе, датотеке са подацима раздвојеним зарезом или табулатором и слично. Стандардни начин приказивања података је табеларни, где колоне представљају променљиве односно атрибуте, а редови представљају појединачне податке, односно сваки ред је један члан поменутог скупа података.

У овом тренутку потребно је креирати скуп података. То је, на пример, могуће учинити коришћењем Excel табеле. Нека та табела приказује податке о групи људи и неким њиховим карактеристикама: табела ће садржати колоне име, пол, висину, тежину, боју очију и боју косе. Нека табела садржи 10 врста то јест податке за 10 људи. Свака врста у овој табели представља један **податак**, односно једну особу у овом конкретном случају. Свака колона представља један **атрибут** података, односно једну карактеристику података.

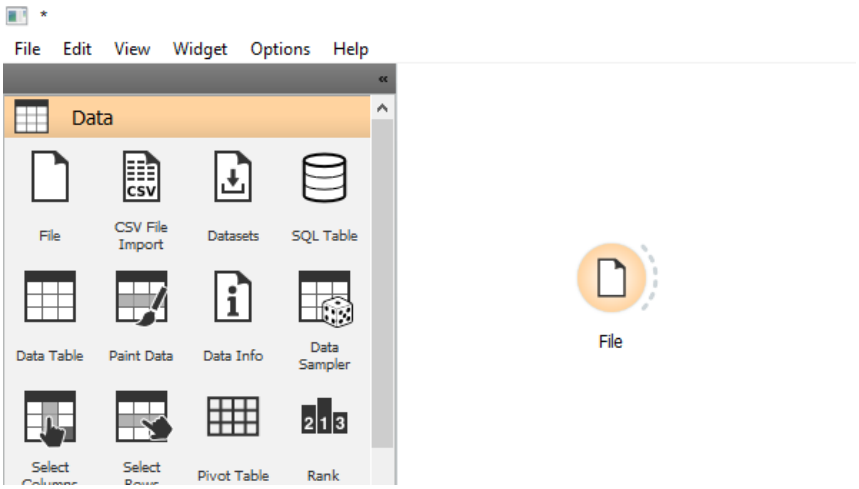


	A	B	C	D	E	F	G	H	I	J	K
1	Ime	Pol	Visina	Težina	Boja očiju	Boja kose					
2	Marko	muški	202	95	plava	crna					
3	Marija	ženski	164	60	braon	crna					
4	Ivana	ženski	159	58	braon	smeđa					
5	Petar	muški	187	82	zelena	plava					
6	Vukašin	muški	192	90	braon	smeđa					
7	Sanja	ženski	170	62	plava	crna					
8	Milica	ženski	175	63	zelena	plava					
9	Jovana	ženski	169	59	braon	crna					
10	Ivan	muški	180	79	braon	smeđa					
11	Strahinja	muški	182	75	plava	smeđa					
12											
13											
14											
15											
16											

Слика 2.1: Скуп података

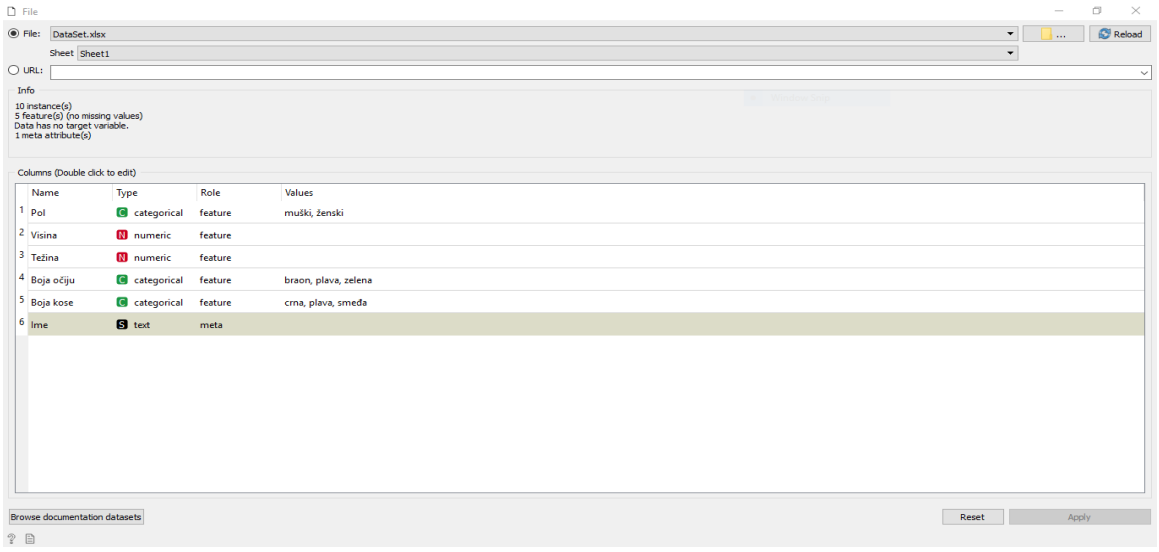
Овај документ је потребно сачувати нпр. под именом *DataSet*. На овај начин креиран је један скуп података представљен *Excel* табелом.

Следећи корак је учитавање овог скупа података у окружење *Orange*. Најпре је потребно креирати нови документ у окружењу *Orange*, и у њега из менија који се налази на левој страни екрана додати оператор *File*. Овај елемент представља опцију да се улазни подаци учитају из неког фајла који се налази на рачунару, или помоћу линка уколико је у питању неки сервис за складиштење података на мрежи. Овако учитане податке касније је могуће представљати на различите начине.



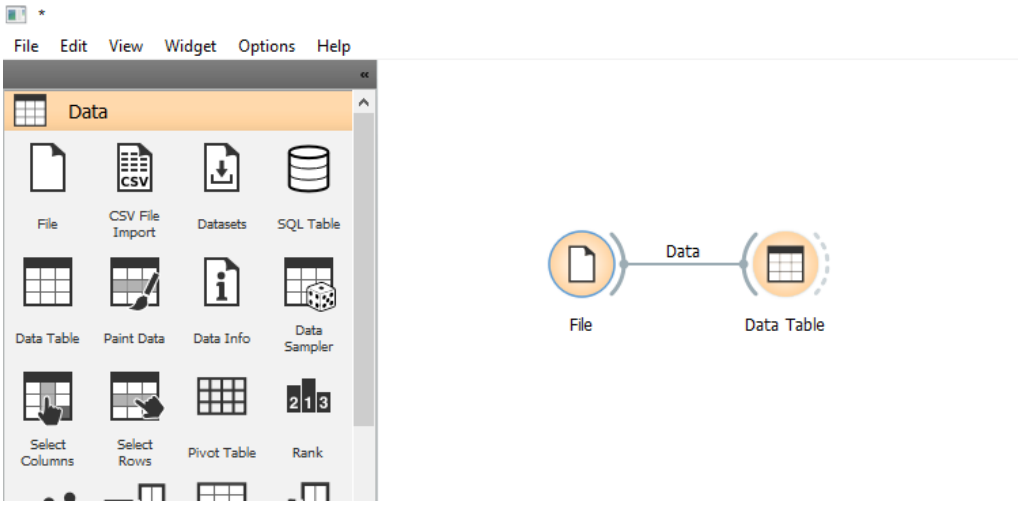
Слика 2.2: Учитавање података

Даље је потребно овај иницијално празан скуп података увезати са жељеним скупом података. У овом случају то ће бити скуп који је претходно направљен. Најпре је потребно кликнути левим тастером миша на оператор *File*, након чега се отвара прозор приказан на слици испод, а затим одабрати жељени документ (уколико се документ налази на рачунару), или налепити одговарајућу везу за случај када је документ складиштен на мрежи, а затим кликнути на дугме *Reset* које се налази у доњем десном углу прозора.



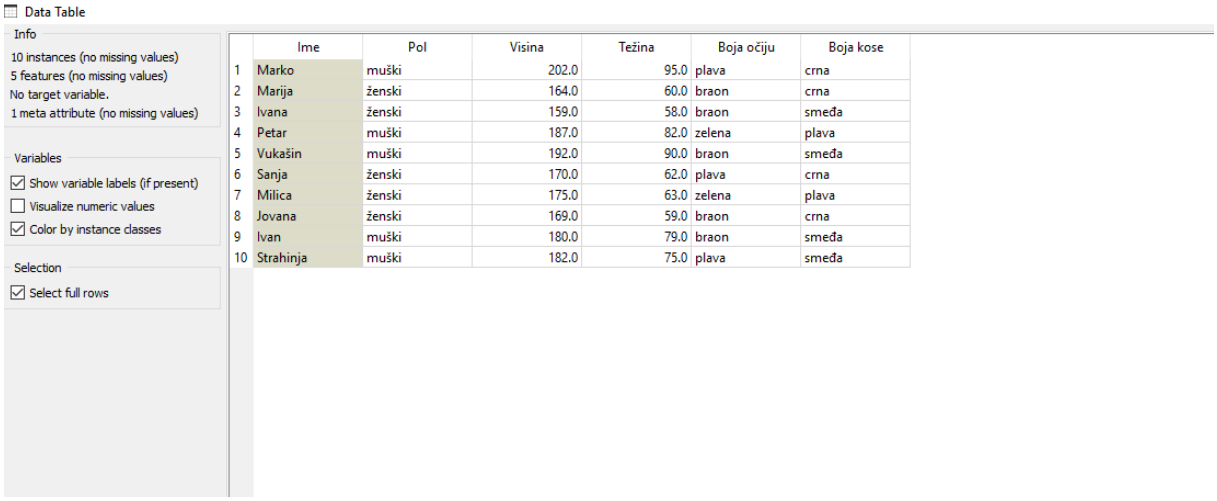
Слика 2.3: Увезивање скупа података

На овај начин је у програмско окружење учитан скуп података. Ради провере да ли је учитан жељени скуп података, могуће је податке приказати у табеларном облику. Потребно је из менија у левом делу прозора одабрати оператор *Data Table*, и повезати га са постојећим фајлом, као што је приказано на слици 2.4.



Слика 2.4: Оператор за табеларни приказ података

Коначно, двокликом на елемент *Data Table* отвара се прозор у коме је приказан садржај табеле. Очигледно, скуп података је правилно учитан.

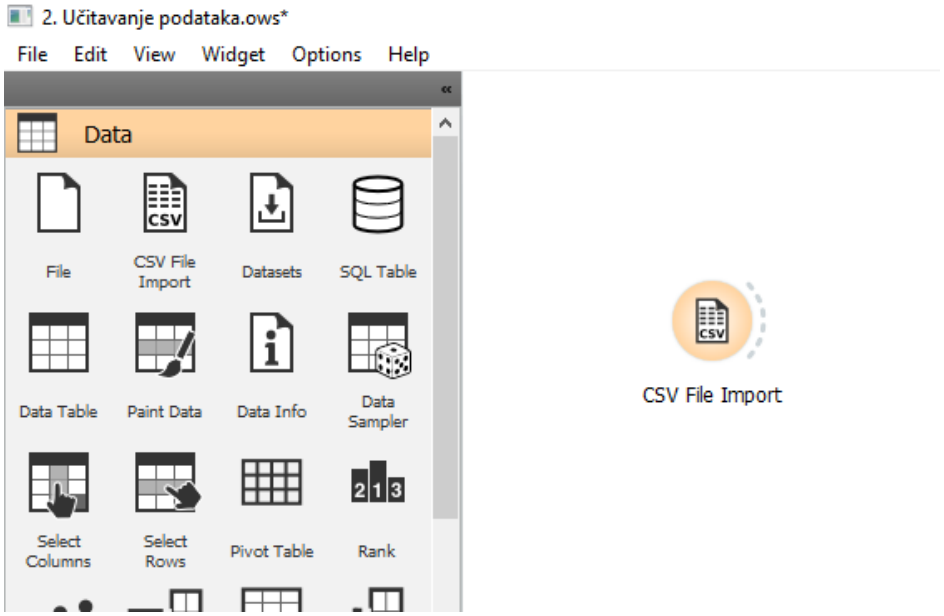


	Ime	Pol	Visina	Težina	Boja očiju	Boja kose
1	Marko	muški	202.0	95.0	plava	crna
2	Marija	ženski	164.0	60.0	braon	crna
3	Ivana	ženski	159.0	58.0	braon	smeđa
4	Petar	muški	187.0	82.0	zelena	plava
5	Vukašin	muški	192.0	90.0	braon	smeđa
6	Sanja	ženski	170.0	62.0	plava	crna
7	Milica	ženski	175.0	63.0	zelena	plava
8	Jovana	ženski	169.0	59.0	braon	crna
9	Ivan	muški	180.0	79.0	braon	smeđa
10	Strahinja	muški	182.0	75.0	plava	smeđa

Слика 2.5: Табеларни приказ скупа података

На потпуно исти начин учитавају се и подаци из датотека са екстензијом *.tab* – екстензија докумената креираних у *Orange* окружењу.

Уколико је потребно учитати податке из документа са екстензијом *.csv*, поступак је аналоган претходно описаном поступку, уз измену да је у првом кораку потребно изабрати оператор *CSV File Import*.



Слика 2.6: Учитавање CSV документа

2.3 Генерализација, потприлагођавање и преприлагођавање и модела

Генерализација (енг. *generalization)* се односи на то колико се добро модел машинског учења показује на новим подцима. Циљ доброг модела машинског учења је добра генерализација од података о тренингу до података на тест скупу, односно података који ће се појавити при примени модела у пракси (које тест скуп и симулира ради давања оцене о квалитету модела). Додатно, основна претпоставка јесте то да подаци из скупа за тренирање и подаци на којима се модел користи припадају истој вероватносној расподели.

Потприлагођавање и преприлагођавање модела су два највећа узрока лоших перформанси метода машинског учења.

Потприлагођавање (енг. *underfitting*) се односи на моделе ограничених капацитета који не могу научити постојеће зависности у оквиру података тренинг скупа, па самим тим ни добро генерализовати на новим подацима. Ова појава је најчешће лако уочљива јер ће модел имати слабе перформансе на подацима за тренинг. Као пример овде се може споменути неуронска мрежа са

једним слојем, која ради са високодимензионалним улазима. Она нема довољно параметара, па ни резултати неће бити задовољавајући.

Преприлагођавање (енг. *overfitting*) се односи на модел који превише добро моделира податке из скупа за тренирање, односно модел који се превише прилагодио подацима. Преприлагођавање се догађа када модел научи детаље приликом обуке на скупу за тренирање до те мере да то негативно утиче на перформансе модела на новим подацима. Модел заправо научи неке зависности и правила која важе само на датом тренинг (или валидационом) скупу, али не важе глобално, што представља проблем када се сусретне са новим подацима. Дакле, модел у том случају одговара одређеном скупу података уз велику прецизност, али зато не може да уклопи додатне податке, или су предвиђања непоуздана на неком новом скупу података.

2.4 Статистике атрибута

Када је реч о припреми и разумевању података и њихових карактеристика, неопходно је осврнути се на неке битне статистике атрибута. Под појмом статистике атрибута подразумевају се најзначајнија својства која карактеришу податке у зависности од њиховог типа.

У случају нумеричких података, то су најчешће минимална вредност, максимална вредност, просечна вредност, најчешћа вредност и стандардна девијација. Стандардна девијација је мера расипања или како су расуте вредности у скупу података. Представља се симболом сигма (σ) и израчунава се као квадратни корен варијансе. Варијанса је просечна вредност квадрата растојања сваке тачке од средње вредности. За разлику од варијансе, стандардна девијација се мери истим јединицама као и полазне вредности (на истој скали).

Када су у питању текстуални или категорички подаци, тада је потребно обратити пажњу на податке о најчешћим вредностима атрибута, најфреквентнијим вредностима атрибута, као и о вредностима атрибута које су јединствене – јављају се само једном и слично. Све ове информације могу бити корисне при обучавању модела, и интензивно се користе у машинском учењу.

Глава 3: Припрема и визуализација података

3.1 Припрема података

У тренутку када је на располагању скуп података, следећи корак, коме многи не придају довољно значаја, јесте припрема података за даљи рад и, што је још важније, њихово разумевање уз помоћ техника визуализације. Овај корак је јако важан за успешност даљег решавања проблема. Алгоритми машинског учења уче из података. Чак и ако су подаци обезбеђени односно доступни, и даље се могу јавити проблеми са њиховим квалитетом, као и неправилностима скривеним у скупу података. Неправилности и нерегуларности су често тешко уочљиве голим оком због количине података у једном скупу. Кључно је обезбедити квалитетне податке за проблем који треба да се реши. Подаци треба да садрже карактеристике које су корисне и значајне за пројекат, али и да буду у адекватном формату.

Избор података у потпуности зависи од проблема који је потребно решити. У машинском учењу, подаци су средство за постизање циља. Другим речима, количина података је важна, али још један јако битан сегмент јесте заправо квалитет података. Постоје неки модели машинског учења који су способни да раде са сировим, неприпремљеним подацима, но далеко је већи број модела који захтевају податке припремљене на прави начин.

Припрема података обично захтева неколико кључних корака.

1. Одабир података: овај корак се односи на одабир подскупа скупа свих доступних података. Другим речима, од свих података који су на располагању потребно је одабрати само оне који су адекватни и релевантни за пројекат. Ово се наравно односи на случајеве када је скуп података такав да се не може цео искористити за тренирање модела. На пример, уколико је на располагању скуп података за учење рукописа који садржи руком писана мала слова, велика слова, бројеве, интерпункцију, а креира се модел за препознавање бројева, одбациће се сви подаци који се не односе на руком писане бројеве.
2. Форматирање података: подаци се могу налазити у различитим датотекама и форматима. Све податке је потребно објединити у једну датотеку чији ће формат бити погодан алгоритму за обраду.
3. „Чишћење података“: у овом кораку потребно је позабавити се недостајућим вредностима и уклонити нежељене вредности из података. Другим речима, у овом кораку потребно је уклонити неке податке и евентуално поправити податке који недостају. Неки подаци могу бити непотпуни, неки могу бити дупликати, а неки просто нису корисни за решавање проблема.
4. Руковање недостајућим подацима: ово је један од најтежих сегмената и онај који ће вероватно трајати најдуже, уколико подаци нису савршени (што је јако ретко случај). Постоји више решења (у складу са типом података) као што су на пример замена недостајућих вредности средњим вредностима, за случај реалних атрибута (атрибути чија је вредност заправо реални број); у случају категоричких атрибута (атрибути који чија вредност може бити једна од неколико фиксираних вредности) може се посегнути за

најфреквентнијом вредношћу, или једноставно обрисати тај податак. Начин решавања овог проблема зависи од самог проблема и типа података.

5. Узорковање: иако ретко, може се десити да има далеко више одабраних података него што је потребно за рад. Више података може резултирати много дужим временом рада алгоритма и већим захтевима у рачунању и меморији. Може се узети мањи репрезентативни узорак одабраних података који могу бити много лакши за визуализацију и прототиповање решења пре разматрања целокупног скупа података.
6. Подела података у скупове за тренинг и тестирање: познато правило поделе података је 80%–20% на скупове за тренирање и тестирање, тим редом (или подела 60% за скуп за тренирање, 20% за скуп за тестирање и 20% за скуп за проверу). Понекад тих 20% података за тест треба да се конструише на начин да нису само случајно издвојени из скупа података (технике стратификације приликом поделе података на подскупове које чувају исту расподелу података у новим скуповима). Важно је напоменути да подаци за тестирање морају бити потпуно независни и не смеју се ни на који начин користити у ранијим фазама.

Претходно наведено илустровано је на следећем примеру. На слици 3.1 приказан је један пример скупа података. Важно је напоменути да је скуп података ових димензија није довољан за коришћење у машинском учењу, те је дат само ради илустрације.

C24

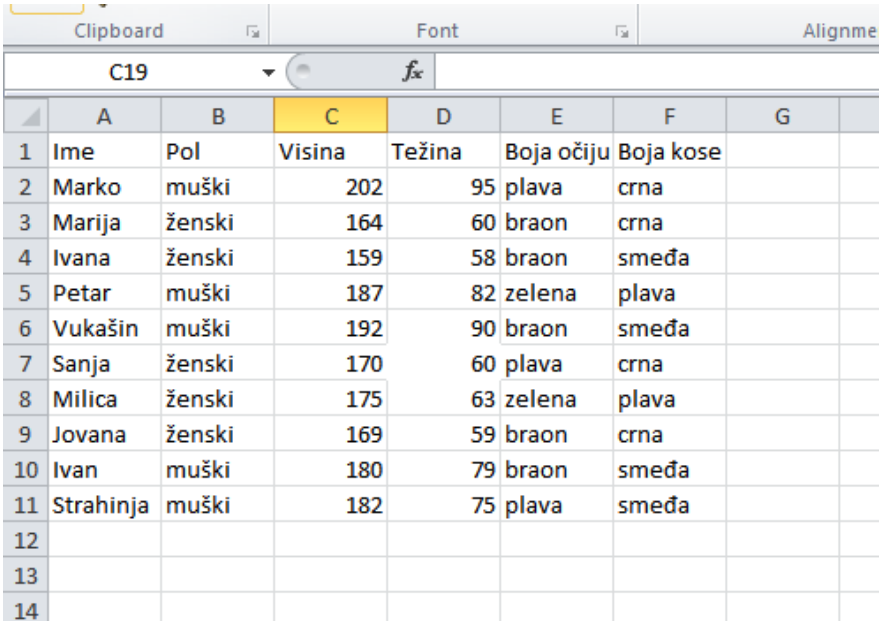
	A	B	C	D	E	F	G
1	Ime	Pol	Visina	Težina	Boja očiju	Boja kose	
2	Marko	muški	202	95	plava	crna	
3	Marija	ženski	164	60	braon	crna	
4	Ivana	ženski	159	58	braon	smeđa	
5	Marko	muški	202	95	plava	crna	
6	Marko	muški	202	95	plava	crna	
7	Petar	muški	187	82	zelena	plava	
8	Vukašin	muški	192	90	braon	smeđa	
9	Sanja	ženski	170		plava	crna	
10	Milica	ženski	175	63	zelena	plava	
11	Jovana	ženski	169	59	braon	crna	
12	Ivan	muški	180	79	braon	smeđa	
13	Strahinja	muški	182	75	plava	smeđa	
14	Nevena	ženski	160	55		crna	
15							
16							
17							
18							

Слика 3.1: Скуп података

Очигледно, у овом скупу података постоје извесни проблеми, и они су обележени жутом бојом. Наиме, Маркови подаци се појављују чак три пута, код Сање не постоји податак о тежини, а такође недостаје податак и о Невениној боји очију.

Да би овај скуп био релевантан, без дилеме је потребно уклонити дупликате. Остаје проблем података који недостају. У случају податка о тежини, сасвим је коректно ово поље попунити просечном тежином свих особа женског пола из овог скупа, што је 60, јер се тиме смањује одступање у подацима, и овакав начин неће негативно утицати на читав скуп података. Остаје још проблем податка о боји очију који недостаје. С обзиром да није могуће логички закључити којом вредности попунити ово поље, најповољније је једноставно обрисати податак. Додатно, како је у питању категоричка вредност, постоји опција да се недостајућа вредност замени најчешће коришћеном вредношћу у скупу података (у овом случају боја очију може се поставити на браон или плаву).

Након ових измена добија се релевантан скуп података без дупликата и недостајућих вредности, што се може видети на слици 3.2.



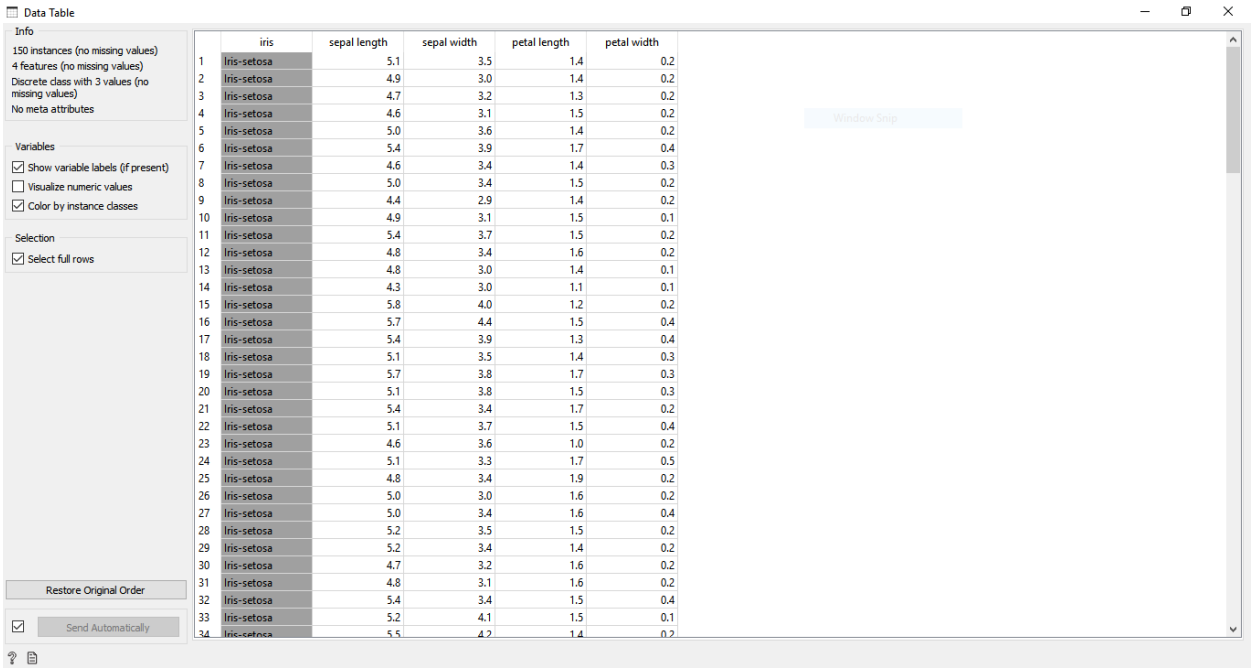
	A	B	C	D	E	F	G
1	Ime	Pol	Visina	Težina	Boja očiju	Boja kose	
2	Marko	muški	202	95	plava	crna	
3	Marija	ženski	164	60	braon	crna	
4	Ivana	ženski	159	58	braon	smeđa	
5	Petar	muški	187	82	zelena	plava	
6	Vukašin	muški	192	90	braon	smeđa	
7	Sanja	ženski	170	60	plava	crna	
8	Milica	ženski	175	63	zelena	plava	
9	Jovana	ženski	169	59	braon	crna	
10	Ivan	muški	180	79	braon	smeđa	
11	Strahinja	muški	182	75	plava	smeđa	
12							
13							
14							

Слика 3.2: Релевантан скуп података

3.2 Визуализација података

У програмском окружењу *Orange* постоји више уграђених опција за визуелно приказивање података односно скупова података. Поред табеларног приказа података који је описан у претходном поглављу, податке је на врло једноставан начин могуће приказати и у облику различитих графикона.

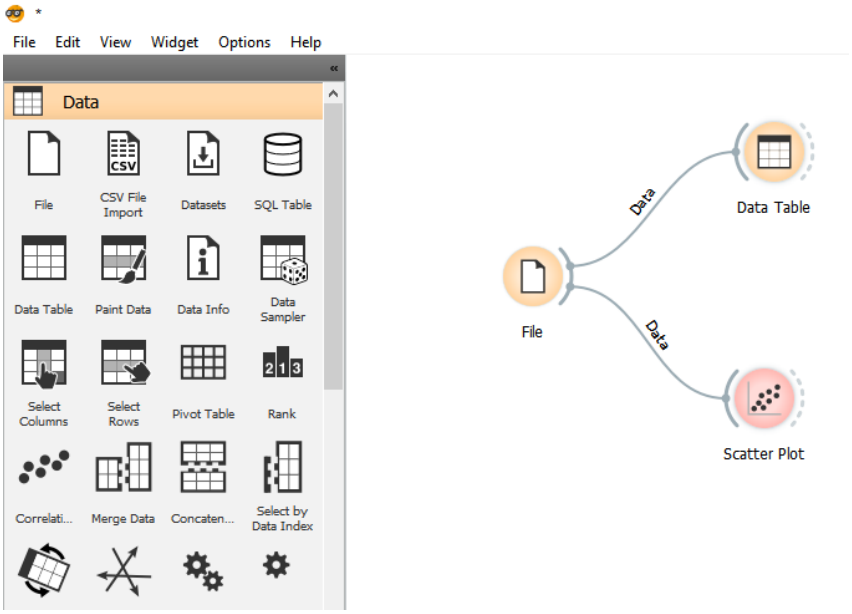
Скуп података *Iris* је већ уграђен у окружење *Orange*. Овај скуп садржи податке о 150 цветова ириса. Цветове описују 4 атрибута: дужина и ширина чашичних листића (*sepal length, sepal width*), и дужина и ширина круничних листића (*petal length, petal width*). Сваком цвету је додељен и атрибут који означава којој, од могуће три врсте ириса (*Iris setosa*, *Iris versicolor*, *Iris virginica*) припада. Табеларни приказ овог скупа података приказан је на следећој слици 3.3.



	iris	sepal length	sepal width	petal length	petal width
1	Iris-setosa	5.1	3.5	1.4	0.2
2	Iris-setosa	4.9	3.0	1.4	0.2
3	Iris-setosa	4.7	3.2	1.3	0.2
4	Iris-setosa	4.6	3.1	1.5	0.2
5	Iris-setosa	5.0	3.6	1.4	0.2
6	Iris-setosa	5.4	3.9	1.7	0.4
7	Iris-setosa	4.6	3.4	1.4	0.3
8	Iris-setosa	5.0	3.4	1.5	0.2
9	Iris-setosa	4.4	2.9	1.4	0.2
10	Iris-setosa	4.9	3.1	1.5	0.1
11	Iris-setosa	5.4	3.7	1.5	0.2
12	Iris-setosa	4.8	3.4	1.6	0.2
13	Iris-setosa	4.8	3.0	1.4	0.1
14	Iris-setosa	4.3	3.0	1.1	0.1
15	Iris-setosa	5.8	4.0	1.2	0.2
16	Iris-setosa	5.7	4.4	1.5	0.4
17	Iris-setosa	5.4	3.9	1.3	0.4
18	Iris-setosa	5.1	3.5	1.4	0.3
19	Iris-setosa	5.7	3.8	1.7	0.3
20	Iris-setosa	5.1	3.8	1.5	0.3
21	Iris-setosa	5.4	3.4	1.7	0.2
22	Iris-setosa	5.1	3.7	1.5	0.4
23	Iris-setosa	4.6	3.6	1.0	0.2
24	Iris-setosa	5.1	3.3	1.7	0.5
25	Iris-setosa	4.8	3.4	1.9	0.2
26	Iris-setosa	5.0	3.0	1.6	0.2
27	Iris-setosa	5.0	3.4	1.6	0.4
28	Iris-setosa	5.2	3.5	1.5	0.2
29	Iris-setosa	5.2	3.4	1.4	0.2
30	Iris-setosa	4.7	3.2	1.6	0.2
31	Iris-setosa	4.8	3.1	1.6	0.2
32	Iris-setosa	5.4	3.4	1.5	0.4
33	Iris-setosa	5.2	4.1	1.5	0.1
34	Iris-setosa	5.5	4.2	1.4	0.2

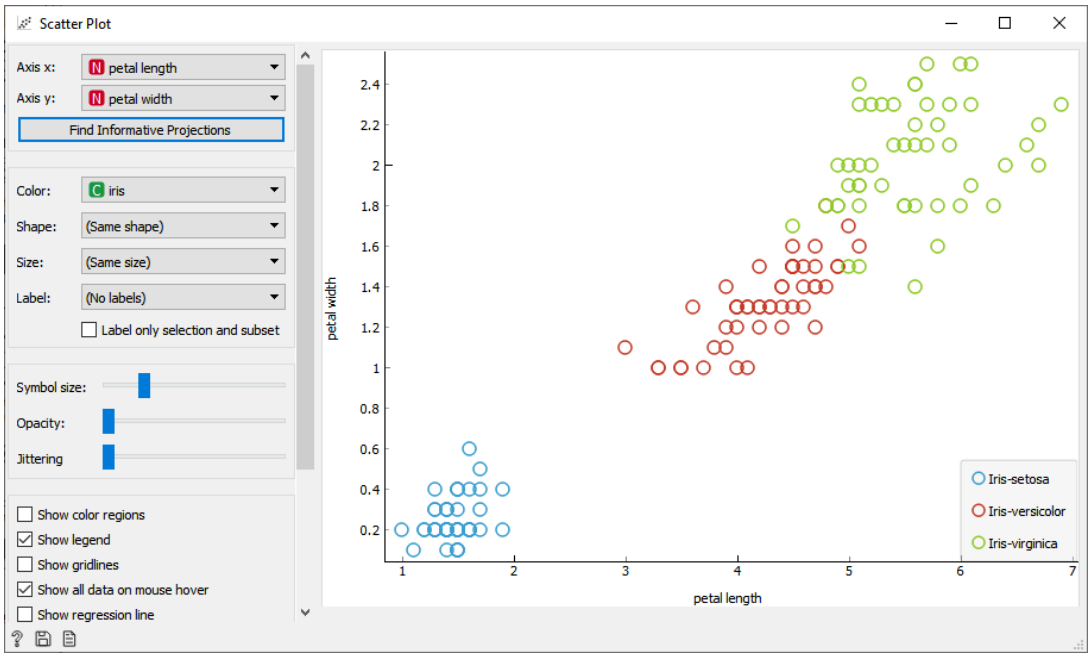
Слика 3.3: Табеларни приказ скупа података *Iris*

Уколико је потребно овај скуп података приказати као тзв. графикон расејања (енг. s*catter plot*), поступак је потпуно аналоган као код табеларног приказа, са изменом да се уместо оператора *Data Table* бира оператор *Scatter plot*. Још је потребно повезати оператор *File* са изабраним оператором за графички приказ података.



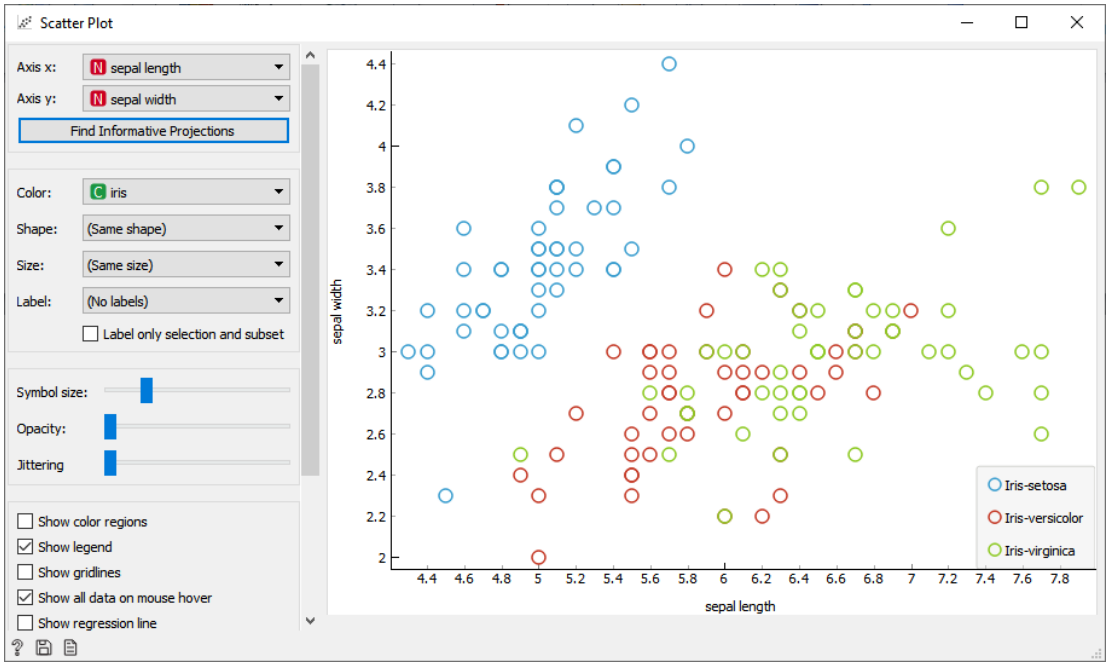
Слика 3.4: Оператор Scatter plot

Двокликом на овај оператор добија se скуп података графички приказан графиконом расејања, као што је приказано на слици 3.5.



Слика 3.5: Графички приказ скупа података Iris коришћењем оператора Scatter plot

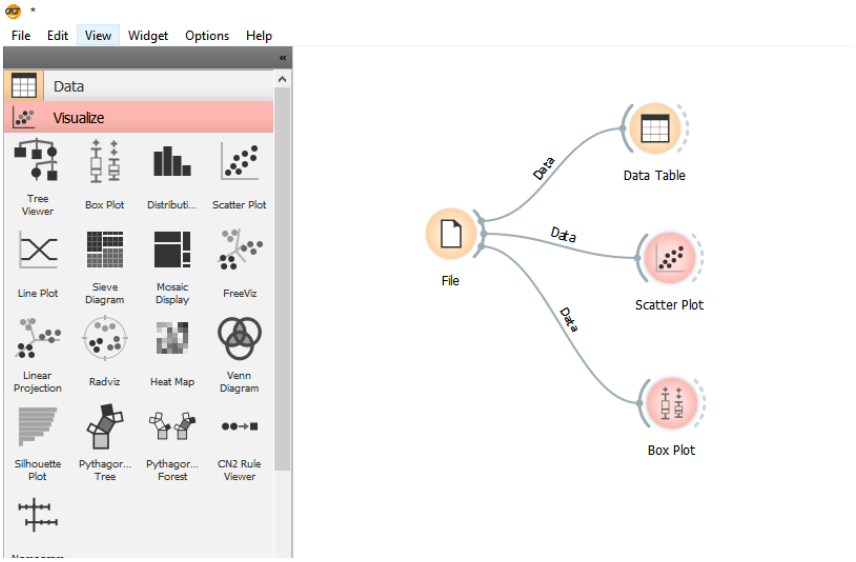
На графику су различитим бојама приказани цветови из три споменуте групе ириса, и то у односу на дужину и ширину круничних листића. Атрибуте који ће се наћи на осама графика могуће је мењати у односу на потребе самог пројекта. На слици је уочљиво да су подаци који припадају различитим класама добро раздвојени (енг. *well separated classes*) у односу на дужину и ширину круничних листића. Уколико се за атрибуте који ће бити приказани на осама графика узму дужина и ширина чашичних листића, то неће бити случај.



Слика 3.6: Графички приказ скупа података Iris коришћењем оператора Scatter plot - 2

Дакле, одабир атрибута који ће бити приказани на осама графика, али и сам начин приказивања, односно тип графика зависи од скупа података, али и потреба пројекта.

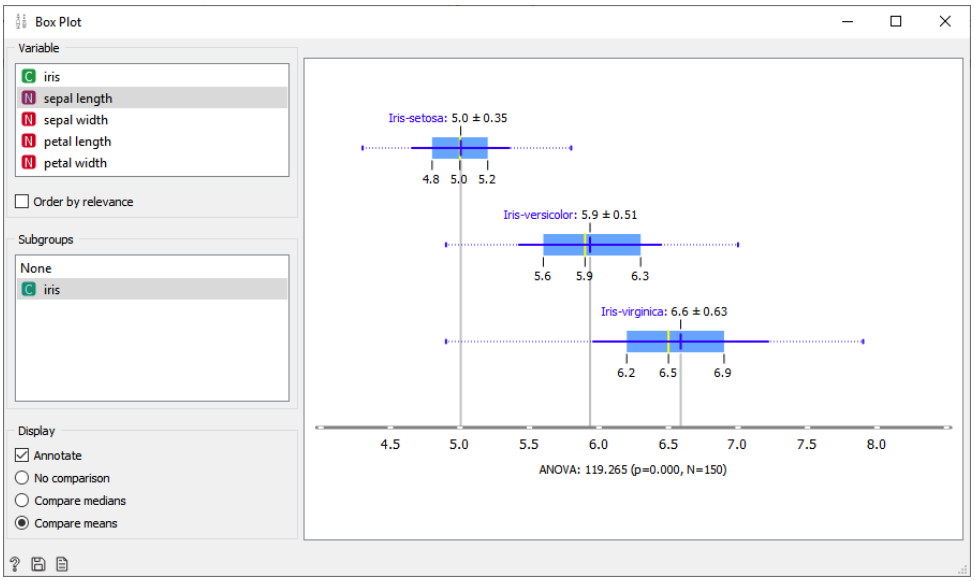
Још један начин приказивања података је *box plot* (слика 3.7).



Слика 3.7: Оператор Box plot

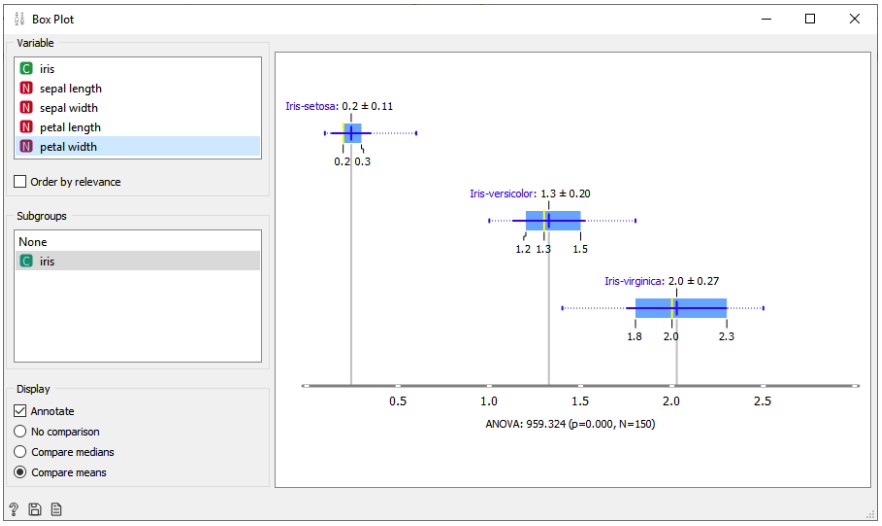
Овај оператор графички приказује распон бројевних вредности одређеног атрибута за сваку од три врсте цветова. Уколико се одаберу различити атрибути који ће бити приказани, и сам графикон ће изгледати потпуно другачије.

На слици 3.8 је приказан график у односу на дужину чашичних листића.



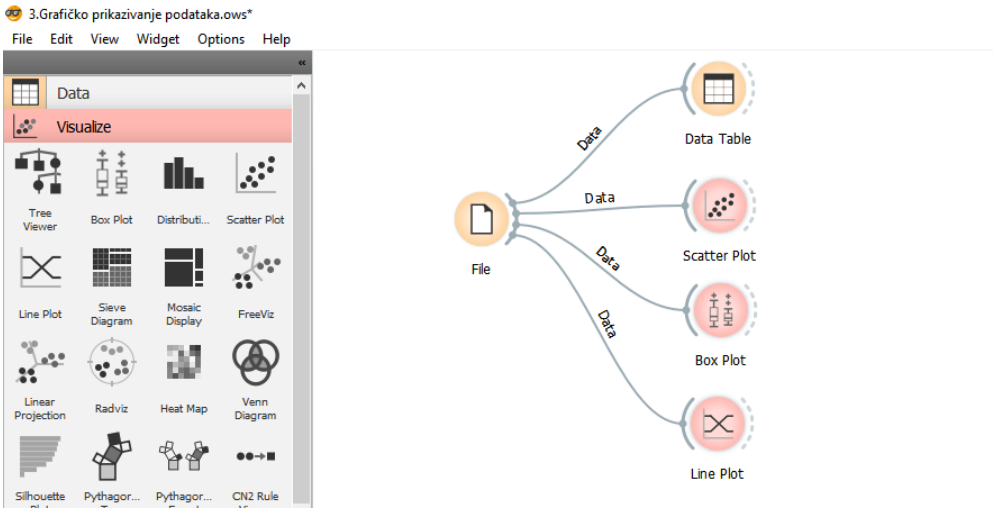
Слика 3.8: Графички приказ скупа података Iris коришћењем оператора Box plot

На слици 3.9 је приказан график у односу на ширину круничних листића:



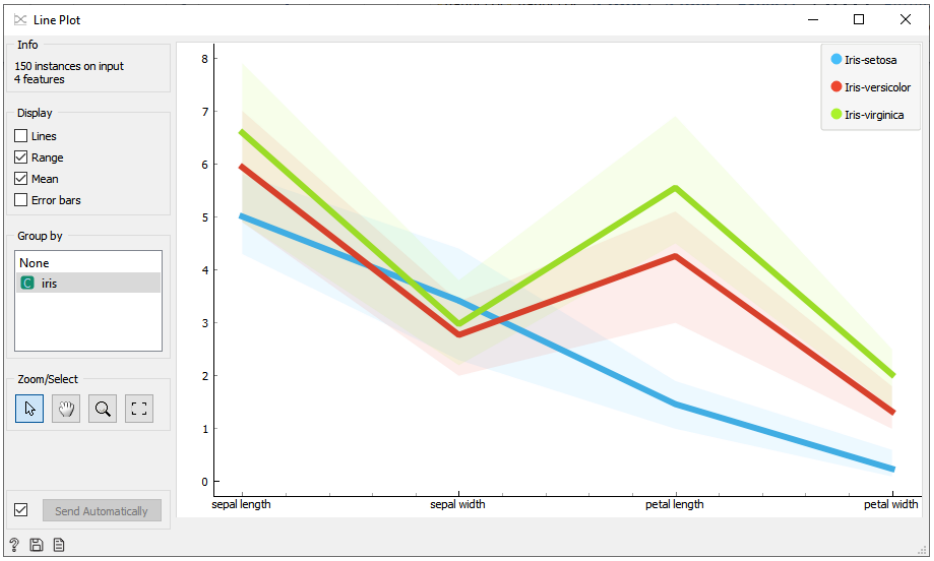
Слика 3.9: Графички приказ скупа података Iris коришћењем оператора Box plot - 2

Још један од начина за графички приказ скупа података је линијски график (енг. *line plot*), што је приказано на слици 3.10.



Слика 3.10: Линеарни график – оператор Line plot

График на слици 3.11. за сваки атрибут упоређује његове просечне вредности за сваку од класа. Приказ и поређење додатних информација за сваки од атрибута могуће је подесити у менију са леве стране.



Слика 3.11: Графички приказ скупа података Iris коришћењем оператора Line plot

Глава 4: Модели помоћу којих се врши анализа података

Алгоритми машинског учења, иако тренутно веома развијени како теоријски тако и практично, и даље напредују и развијају се. У већини случајева, као што је већ речено, алгоритми који се користе у машинском учењу грубо се разврставају у три категорије: надгледано учење, ненадгледано учење и учење поткрепљивањем.

Код надгледаног учења скуп података укључује и жељене излазе (ознаке за задате податке из скупа података - њих називамо означеним) тако да функција може израчунати грешку за дато предвиђање која представља меру одступања од задатих ознака. Ово значи да у скупу података већ постоје информације којој класи дати податак припада (у случају класификације), или је позната нумеричка вредност циљног атрибута (у случају регресије). Када се изврши предвиђање и добије грешка, модел се у односу на то, у фази тренирања или валидације може изменити ради бољих резултата. Већина модела машинског учења своје учење заснива на томе да покушава да минимизује вредност грешке. У надгледаном машинском учењу постоје две основне врсте проблема: регресија и класификација. **Регресија** представља проблем предвиђања непрекидне циљне променљиве. **Класификација** је заправо разврставање података у класе, односно предвиђање категоричке циљне вредности.

Код ненадгледаног учења скуп података не укључује жељени излаз. Уместо тога модел покушава извести одређене закључке, односно уочити неку врсту структуре која је присутна на основу података који су на располагању. Другим речима, метод покушава да уочи неку зависност између атрибута односно особина датих података, и да их на основу тога групише у класе са сличним особинама (у случају кластеровања, односно груписања података у кластере).

Учење поткрепљивањем се најчешће користи када је реч о проблемима које је потребно решити предузимањем низа акција (као што је то у случају самовозећих аутомобила, када је потребно превести возило од тачке А до тачке Б предузимањем низа радњи – притискања гаса и кочнице, померањем волана и слично). Потребно је научити модел, да у зависности од стања у коме се тренутно налази, као и од опаженог стања околине предузима краткорочне акције које воде ка испуњењу дугорочног циља (у зависности од брзине којом се креће, као и од стања на улици – број возила, временски услови, самовозећи аутомобил треба прилагодити своју брзину кретања).

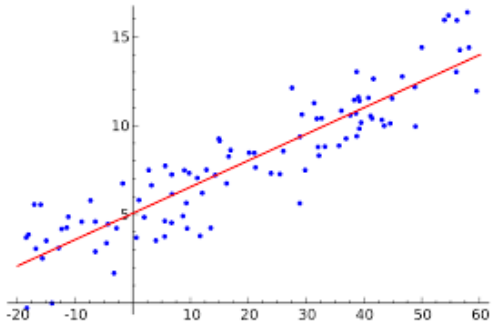
Даље ће бити дат кратак приказ кључних метода сваке од наведених категорија.

4.1 Модели надгледаног учења

Модели односно алгоритми који се најчешће користе у случају надгледаног учења могу се поделити у две основне категорије: регресија и класификација. Као што је малопре речено, код регресионих модела излаз (циљна променљива односно вредност која се предвиђа) је непрекидна променљива, док је код класификационих модела излаз дискретан. У случају класификације може постојати две или више класа.

4.1.1 Линеарна регресија (енг. *linear reggression*)

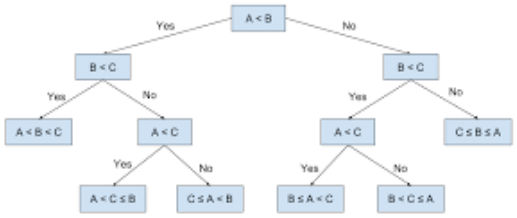
Линеарна регресија представља један од најједноставнијих и најчешће коришћених модела машинског учења. Она на једноставан начин проналази линеарну функцију која најбоље апроксимира податке. Линеарност се овде односи на линеарност по параметрима, тако да функција може бити и полиномијална. Пример графика линеарне регресије је приказан на слици 4.1.



Слика 4.1: График линеарне регресије

4.1.2 Стабло одлучивања (енг. *decision tree*)

Стабло одлучивања може представљати како класификациони тако и регресиони модел надгледаног учења. Алгоритми овог типа стварају стабла која предвиђају резултат улазног вектора на основу правила одлучивања изведених из карактеристика присутних у подацима. Стабло одлучивања је користан модел у машинском учењу јер га је лако визуализовати, тако да је могуће разумети кораке који доводе до резултата. Једноставније речено, код овог модела је јасно на основу чега модел доноси закључке, што се може видети на слици 4.2. Изражена интерпретабилност једна је од значајнијих предности овог модела.

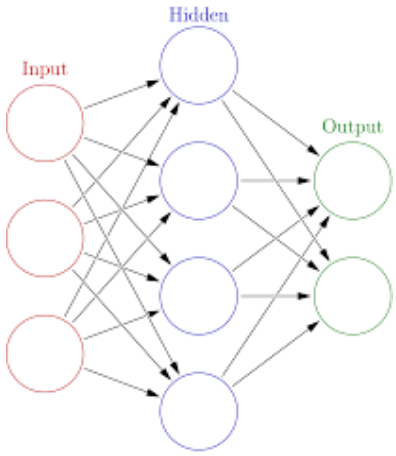


Слика 4.2: Стабло одлучивања

Стабло одлучивања је дијаграм налик структури у којој сваки унутрашњи чвор представља "тест" вредности појединачног атрибута, свака грана представља исход теста и сваки лист представља класу ознаке (одлука донета после рачунања свих атрибута). Стазе од корена до листа представљају правила класификације. У случају регресије, модификација је та што листови стабла не чувају категоричку, већ нумеричку вредност.

4.1.3 Неуронске мреже (енг. *neural network*)

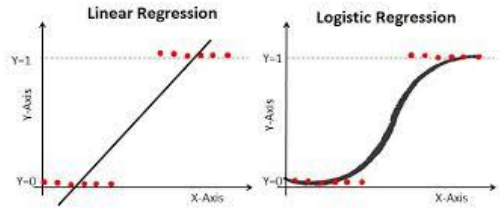
Неуронска мрежа представља модел машинског учења који је биолошки инспирисан људским мозгом (везе између неурона, аксона, дендрита и начин функционисања људског мозга уопште покушава се пресликати у математичке моделе учења). Постоји већи број подврста неуронских мрежа од којих потпуно повезане неуронске мреже (енг. *feed-forward neural networks*) представљају најједноставнију варијанту. Модел потпуно повезане неуронске мреже састоји се од неурона који су организовани у слојеве, при чему су неурони из суседних слојева међусобно повезани. У случају већег броја слојева мрежа се назива дубоком, а унутрашњи слојеви скривеним слојевима. Илустрација изгледа неуронске мреже са једним скривеним слојем приказана је на слици 4.3.



Слика 4.3: Неуронска мрежа

4.1.4 Логистичка регресија (енг. *logistic regression*)

Логистичка регресија, упркос свом имену, представља модел класификације. Она се користи за моделирање вероватноће коначног броја исхода, обично два. У суштини, логистичка регресија је креирана на такав начин да излазне вредности могу бити само између 0 и 1. У статистици се логистички модел користи за моделирање вероватноће одређене класе или догађаја који постоје, попут пролаза/неуспеха, победе/губитка, живог/мртвог или здравог/болесног.

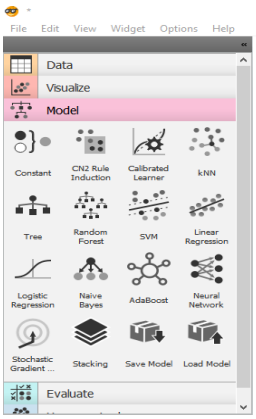


Слика 4.4: Линеарна и логистичка регресија

4.1.5 Метода потпорних вектора (енг. *support vector machine – SVM*)

Метода потпорних вектора је модел надгледаног учења који може постати математички прилично компликован уколико се зађе у детаље, али је прилично интуитиван на најосновнијем нивоу. Уз претпоставку да постоје две класе података, ова метода ће наћи границу између две класе података односно хиперраван која на најбољи могући начин раздваја податке који припадају различитим класама. Нови податак се класификује у зависности од тога са које стране хиперравни се налази. Другим речима, модел тежи да нађе хиперраван која најбоље раздваја податке две класе, односно која је на највећем могућем растојању од потпорних вектора.

Сви ови модели, али и неки додатни доступни су у окружењу *Orange,* у менију који се налази левој страни прозора, у делу под називом *Model*, који је приказан на слици 4.5.



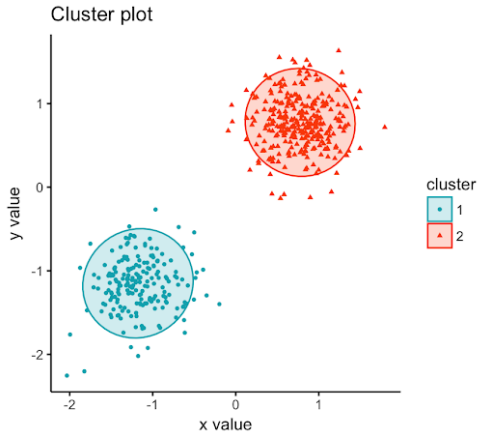
Слика 4.5: Модели машинског учења у окружењу Orange

4.2 Модели ненадгледаног учења

4.2.1 Кластеровање (енг. *clustering*)

Кластеровање је техника ненадгледаног учења која подразумева груписање (кластеровање) података.

Уобичајене технике кластеровања су **кластеровање к-средина** (енг. *K-means clustering*), **хијерархијско кластеровање** (енг. *hierarhical clustering*), и **кластеровање на основу густине** (енг. *density-based spatial clustering of applications with noise - DBSCAN*). Иако свака техника има различиту методу проналажења кластера, све имају за циљ постићи исту ствар – груписање сличних података у кластере то јест групе, што је илустровано на слици 4.6.

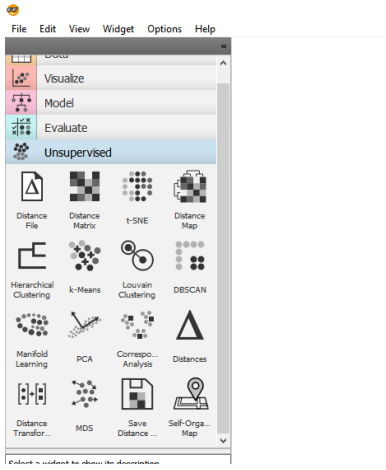


Слика 4.6: Кластеровање

4.2.2 Анализа главних компоненти (енг. *principal component analysis PCA*)

У најједноставнијем смислу ова метода подразумева да се подаци већих димензија сведу на мањи простор (нпр. 2 димензије). То резултира нижом димензијом података, који су лакши за обраду (рад са подацима великих димензија може бити проблематичан, како из рачунских тако и из разлога визуелизације и интуитивне представе података). На овај начин се мења и координатни систем, и сами атрибути. Ово је метода редукције димензионалности података и често се користи пре неког другог модела надгледаног учења.

Поменути модели, али и неки додатни доступни су у окружењу *Orange*, на левој страни прозора, у делу под називом *Unsupervised*, који је приказан на слици 4.7.



Слика 4.7: Модели машинског учења у окружењу Orange

Глава 5: Евалуација (оцењивање) модела

За сваки проблем машинског учења неоспорно постоји више метода које се могу користити за његово решавање. За сваки метод постоји бесконачно много модела који се могу користити, а који се добијају мењањем неког од параметара. Тренирање модела се заснива на минимизовању грешке подешавањем параметара модела (а параметри су ништа друго до реални бројеви) да би модел достигао највећу могућу тачност приликом решавања конкретног проблема. Намеће се питање како изабрати који модел је најпогоднији за задати проблем? Ово се решава евалуацијом модела.

Дакле, сви модели доступни за решавање проблема се морају на неки начин оценити, како би се изабрао најбољи. **Евалуација модела** јесте оцењивање квалитета модела. Неопходно је знати да ли су предвиђања која модел направи заиста у довољној мери прецизна и, сходно томе, може ли се веровати његовим предвиђањима. Често се овај сегмент машинског учења неправедно запоставља, не посвећује му се довољно пажње и не приступа му се са довољно озбиљности, што представља велику грешку, јер од избора модела умногоме зависи успешност целог пројекта.

Мере квалитета модела варирају у зависности од типа проблема који се решава (нпр. да ли је у питању регресија, класификација или кластеровање), жељене поузданости оцене, количине података и својстава метода чији се модели евалуирају.

5.1 Подела скупа података и важност ове поделе

Најбитнија ствар на коју треба обратити пажњу да би се правилно оценио квалитет модела, а уједно и главно начело евалуације модела јесте подела доступног скупа података. Подаци коришћени при тренирању модела не смеју бити кришћени приликом евалуације. Дакле, погрешно је тренирати модел на читавом скупу података, или чак на било који начин користити податке за тестирање у фази тренирања модела. Јако је важно користити нове податке приликом евалуације модела како би се смањила вероватноћа преприлагођавања скупу за тренирање (енг. *overfitting*). Преприлагођавање је јако чест проблем у машинском учењу и представља ситуацију при којој модел одлично моделује податке који се налазе у скупу за тренирање, али зато показује јако лоше перформансе приликом евалуације на скупу за тестирање, односно новим подацима који му нису били доступни у фази тренинга. Дакле, у овом случају модел је научио и неке зависности које важе само на скупу за тренирање, а које се не могу пресликати на нове податке и не представљају суштинске везе које важе глобално, на свим подацима из расподеле која се моделује. Ово се може илустровати на примеру ученика који одређену лекцију научи напамет. Дакле, он ту лекцију зна потпуно прецизно („скуп података за тренирање“), али није научио и разумео зависности и правила која се односе на ту област, па ће његово знање на било ком сличном градиву („скуп података за тестирање“) бити јако лоше.

Типично раздвајање података на тренинг податке и тест податке било би коришћење 80% података за тренирање и 20% података за тестирање. Наравно, оваква подела није обавезна, нпр. некада се 70% података користи за тренирање а 30% података за тестирање, и слично

Параметри модела уче се на тренинг скупу. У случају када је потребно одредити најбољу кнфигурацију модела као подскуп скупа података за тренинг издваја се скуп за валидацију. На овом скупу је могуће одредити најбољу архитектуру неуронске мреже или максималну дубину стабла одлучивања, такве да су перформансе модела трениранога са том конфигурацијом на тренинг скупу најбоље. У овом случају уобичајено би било да 60% података буде у скупу за тренинг, 20% података у скупу за валидацију и 20% података у скупу за тестирање, мада се могу користити и другачије поделе скупа података.

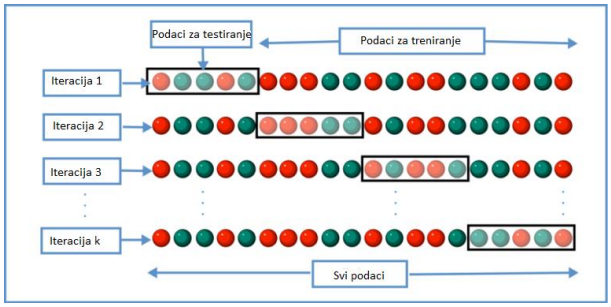
5.2 Унакрсна валидација

К-слојна унакрсна валидација представља технику евалуације модела при којој је оригинални скуп података подељен на *к* скупова једнаке величине (тзв. слојеви). Од тих *к* подскупова, *к-1* подскупова се користи као скуп за тренинг, док један подскуп служи као скуп за тестирање. Овај поступак поделе и тестирања се понавља *к* пута, где је *к* број који бира сам корисник, најчешће 5 или 10. Процена грешке се добија као просек грешака из ових *к* испитивања, и тако се добија ефикасност датог модела.

На пример, када се врши петострука унакрсна валидација, подаци се прво деле на 5 делова (приближно) једнаке величине. Тренира се низ модела. Први модел се тренира користећи први подскуп као тест скуп, а преостали подскупови се користе као скуп за тренирање. То се понавља за сваки од постојећих 5 делова података. Нека је, на пример, у питању проблем класификације, и нека је мера квалитета модела која се посматра прецизност предвиђања. Процена прецизности модела добија се на основу просека процена за свих 5 испитивања.

Као што се види, свака тачка података мора бити у тестном скупу тачно једном и мора бити у скупу за тренинг *к-1* пута. Ово значајно смањује диспрезију оцене и повећава ефикасност ове методе.

Техника К-слојне унакрсне валидације је илустрована на слици 5.1.



Слика 5.1: Унакрсна валидација

Глава 6: Мере квалитета модела

6.1 Мере квалитета модела код класификације

Мере квалитета модела користе се за оцењивање у којој мери је неки модел добар или лош. Већина мера квалитета које се користе код класификације заснивају се на **матрици конфузије** (енг. *confusion matrix*). Матрица конфузије даје детаљнију слику тачних и погрешних класификација за сваку класу. Број тачних и нетачних предвиђања укршта се са вредностима предвиђања и рашчлањује се по категоријама. Она не само да даје увид у грешке класификатора, већ што је још важније стиче се утисак о врстама грешака које се чине.

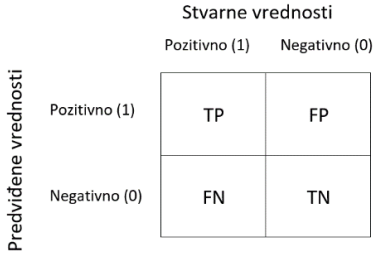
Матрица конфузије је матрица $M = [m_{ij}]$ елемент m_{ij} означава број елемената класе i који су класификовани у класу j. У случају када је матрица конфузије дијагонална, класификација је потпуно тачна, односно модел класификује податке без грешке. Елементи матрице конфузије који се не налазе на главној дијагонали означавају тачно класификоване податке.

Уколико се посматра проблем бинарне класификације, тада постоје две класе, и најчешће се једна класа назива позитивном, а друга негативном класом. Тада постоје четири поља у матрици конфузије, и сваки податак припада тачно једној од њих.

Дефиниција поља у матрици конфузије за случај бинарне класификације:

- стварно позитивна класа (енг. *true positive - TP*): елементи скупа података који су обележени као позитивни и од стране модела препознати као позитивни:
- лажно негативна класа (енг. *false negative - FN*): елементи скупа података који су обележени као позитивни а од стране модела проглашени негативним;
- стварно негативна класа (енг. *true negative - TN*): елементи скупа података који су обележени као негативни и од стране модела препознати као негативни;
- лажно позитивна класа (енг. *false positive - FP*): елементи скупа података који су обележени као негативни а од стране модела проглашени позитивним.

Тада матрица конфузије има следећи облик:



Слика 6.1: Матрица конфузије за проблем бинарне класификације

Уколико постоји више од две класе којима могу припадати подаци, и матрица конфузије је веће димензије (при чему је у том случају потребно одустати од позитивних и негативних ознака).

Мере квалитета модела које се најчешће користе код проблема класификације су површина испод ROC криве (енг. *receiver operating characteristic curve*), лифт крива (енг. *the lift curve*), тачност класификације (енг. *classification accuracy*), прецизност и одзив (енг. *precision and recall*), и F1 мера.

1. **Тачност класификације** (енг. *classification accuracy*)

Тачност класификације као најинтуитивнија мера у оцени класификационих модела говори да ли се модел правилно тренира и како уопште може функционисати.

Тачност класификације представља удео исправно класификованих података у укупном броју података. Формула за израчунавање тачности класификације код проблема бинарне класификације је дата у наставку:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN}.$$

Иако интуитивна, ова мера не даје детаљне информације о квалитету модела који се оцењује. Проблем са коришћењем тачности као главне метрике је тај што не даје добру информацију када постоји озбиљна неравнотежа класа (уколико на пример у једној класи има знатно више података него у другој). Ово се може видети на примеру детекције терориста. Од свих људи који се разматрају, само мали број њих припада класи терориста. Уколико модел све окарактерише као да нису терористи, прецизност ће заиста бити велика (јер велики проценат људи заправо и нису терористи), али је овај модел онда потпуно бескористан.

2. **Прецизност и одзив** (енг. *precision and recall*)

Прецизност је удео позитивних података у свим подацима који су проглашени позитивним. Она помаже у случајевима када је удео лажних позитивних резултата велики. У наставку следи формула којом се израчунава прецизност:

$$Prec = \frac{TP}{TP+FP}.$$

Одзив је удео пронађених позитивних података у свим позитивним подацима. Супротно од прецизности, одзив помаже када је удео лажних негативних резултата велики. Формула којом се израчунава одзив је:

$$Rec = \frac{TP}{TP+FN}.$$

Прецизност и одзив су мере квалитета модела које када се посматрају одвојено не дају информацију која може бити претерано корисна, зато се најчешће користе заједно.

3. **F_1 мера**

F_1 мера је свеобухватна мера прецизности модела која комбинује прецизност и одзив (она је њихова хармонијска средина). Другим речима, добар резултат F_1 мере значи да је удео лажно позитивних и лажно негативних резултата низак. F_1 мера се израчунава по следећој формули:

$$F_1 = 2 \cdot \frac{Prec \cdot Rec}{Prec + Rec}.$$

Када резултат F_1 мере има већу вредност, односно вредност близу 1, модел се сматра добрим, док је модел лоше оцењен када резултат F_1 мере има мању вредност, односно вредност близу 0.

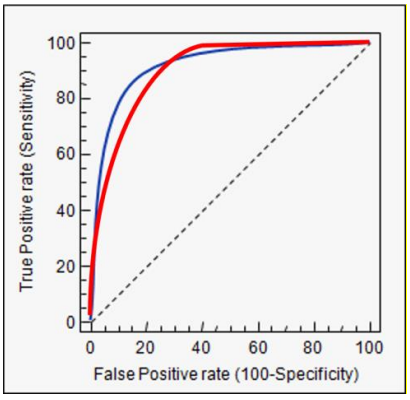
4. **ROC крива** (енг. *ROC (receiver operating characteristic) curve*)

Када је потребно проверити или визуелизовати перформансе модела којим се решава проблем класификације у две класе, користи се AUC (енг. *AUC: area under the ROC curve*). Ово је је једна од најважнијих техника за оцењивање перформанси бинарних класификационих модела са небалансираним класама.

AUC представља степен или меру раздвајања, и то је реалан број између 0 и 1. Овај број говори колико је модел способан да разликује класе. Одличан модел има AUC близу 1, што значи да има добру меру одвојивости. Лош модел има AUC близу 0.5, што интерпретирамо као немогућност модела да на прави начин раздвоји податке две класе. ROC крива је график који приказује перформансе класификацијског модела на свим нивоима класификације. Ова крива приказује два параметра:

- стопа стварно позитивних (TPR): $TPR = \frac{TP}{TP+FN}$.
- стопа лажно позитивних (FPR): $FPR = \frac{FP}{FP+NT}$.

ROC крива црта ове две променљиве - TPR у односу на FPR при различитим праговима класификације. Спуштањем прага класификације више ставки се класификује као позитивно, чиме се повећавају и лажни и стварни позитивни.

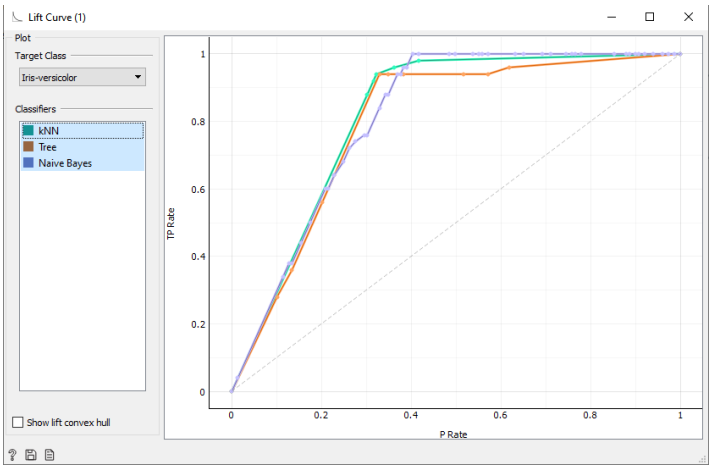


Слика 6.2: ROC крива

5. **Лифт крива** (енг. *the lift curve*)

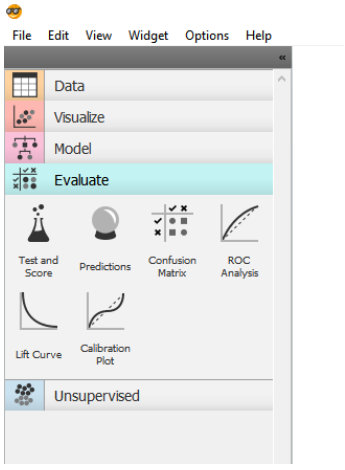
Лифт крива приказује однос између података класификованих као позитивних и укупног броја позитивних података, и на тај начин мери перформансе изабраног класификатора. На x −оси налази се популација (енг. *P Rate*), док се на y −оси криве налази TPR, односно стопа стварно позитивних (6.1). Класификатор има боље перформансе уколико је површина испод лифт криве већа.

У случају који је приказан на слици 6.3 приказана је лифт крива за проблем класификације на скупу података *Iris*, при чему су коришћене три методе класификације – стабло одлучивања, наивни Бајесов алгоритам и метод k −најближих суседа. Дакле, најбоље оцењен класификатор је наивни Бајесов алгоритам, затим метод k −најближих суседа и на крају стабло одлучивања.



Слика 6.3: Лифт крива

Поменуте мере квалитета модела, али и неке додатне доступне су у окружењу *Orange*, на левој страни прозора, у делу под називом *Evaluate* (приказано на слици 6.4), и биће приказане у поглављима која следе.



Слика 6.4: Мере квалитета модела у окружењу Orange

6.2 Мере квалитета модела код регресије

Када је реч о проблему регресије, мере квалитета модела које се најчешће користе су средњеквадратна грешка и њен корен (енг. *mean square error, root mean square error*) и коефицијент детерминације R^2.

1. **Средњеквадратна грешка** и **корен средњеквадратне грешке** (енг. *mean square error MSE, root mean square error)*

Средњеквадратна грешка процењује очекивану квадрирану разлику између моделом предвиђених и стварних вредности циљне променљиве. То је збир, у свим тачкама података, квадрата разлике између предвиђених и стварних вредности циљних променљивих, подељен са бројем података. Средњеквадратна грешка се рачуна по формули:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - {y'}_i\right)^2,$$

где је n – број тачака података, y_i – предвиђена вредност променљиве за дату тачку података, а ${y'}_i$ – стварна вредност променљиве у тој тачки.

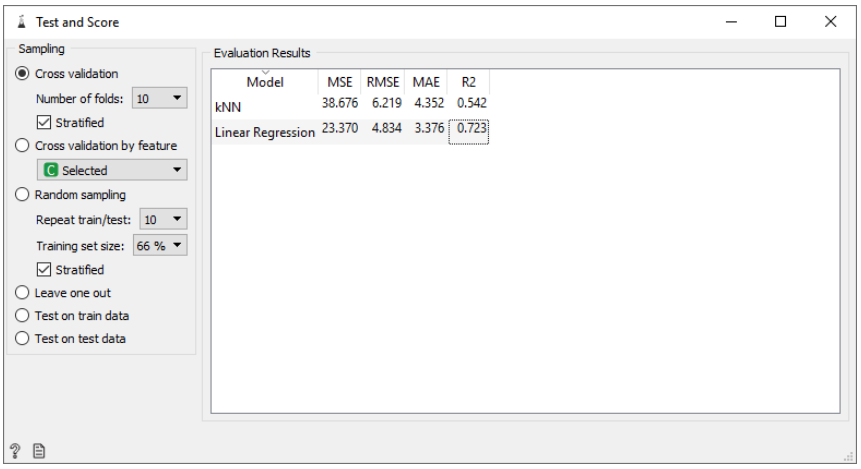
Корен средњеквадратне грешке представљен је кореном ове вредности. Његова предност у односу на средњеквадратну грешку из које се и изводи, јесте чињеница да је мерен на истој скали као и циљна променљива.

2. **Коефицијент детерминације R^2**

Коефицијент детерминације интуитивно представља проценат варирања циљне променљиве који је успешно описан креираним моделом. Коефицијент детерминације је са горње стране ограничен бројем 1.

Ако је коефицијент детерминације 0, то значи да се зависна промељива не може предвидети из независне променљиве. Уколико је коефицијент детерминације једнак 1, то значи да зависна променљива може да се предвиди без грешке из независне променљиве. Када је коефицијент детерминације између 0 и 1, он указује на степен до којег је зависну варијаблу могуће предвидети. Додатно, коефицијент детерминације може бити и негативан јер натренирани модел који се оцењује може бити произвољно лош.

Наведене мере квалитета код проблема регресије се у програмском окружењу *Orange* могу видети кликом на оператор *Test and Score* (слика 6.5).



Слика 6.5: Мере квалитета модела

Глава 7: Класификација и регресија

Класификација је, као што је већ речено, представља разврставање података у класе. Бинарни класификатори раде са само две класе односно два могућа исхода (на пример: позитиван или негативан одговор на неко питање; да ли пацијент има неку болест или не и слично). Постоје и вишекласни класификатори који задате податке сврставају у више класа (на пример: којој земљи припада застава, да ли је на слици ружа, лала или орхидеја итд). У овом случају се претпоставља да је сваки узорак додељен само једној класи. Постоје и варијације код којих један податак припада већем броју класа, али о њима неће бити речи.

Неки практични примери проблема класификације су: препознавање говора, препознавање рукописа, препознавање лица, биометријска идентификација, класификација докумената, класификација тумора итд.

Постоји неколико врста метода за класификацију у машинском учењу, као што су:

- логистичка регресија (енг. *logistic regression*);
- наивни Бајесов класификатор (енг. *naive Bayes classifier*);
- метод К најближих суседа (енг. *kNN – K nearest neighbours*);
- метод потпорних вектора (енг. *SVM – support vector machine*);
- стабла одлучивања (енг. *decision tree*);
- метод случајних шума (енг. *random forest*);
- неуронске мреже (енг. *neural network*).

Регресија је тип проблема се користи за предвиђање реалне вредности циљне променљиве. Један од најчешће коришћених и најједноставнијих примера регресије јесте предвиђање цене куће на основу неких карактеристика као што су њена локација, број соба и слично (скуп података под називом *Housing*).

Неке од важнијих врста регресијскионих метода у машинском учењу:

- линеарна регресија (енг. *linear reggression*);
- метод *k* најближих суседа (енг. *kNN – K nearest neighbours*);
- стабла одлучивања (енг. *decision tree*);
- метод случајних шума (енг. *random forest*);
- неуронске мреже (енг. *neural network*).

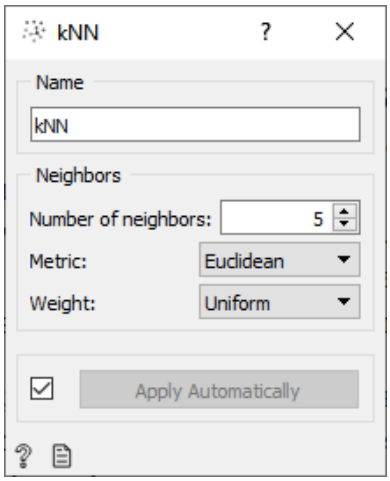
Важно је напоменути да се неки модели, попут неуронских мрежа или технике најближих суседа, уз мале модификације могу користити како за проблеме регресије тако и за проблеме класификације.

7.1 Метод к-најближих суседа

Метод к-најближих суседа (енг. *K-nearest neighbor, KNN*) је често коришћен, јако интуитиван метод машинског учења. Може се користити за класификацију са произвољним бројем класа. Ово је једноставан метод који чува све доступне податке при чему нове податке класификује на основу сличности са подацима из скупа за тренирање (на основу такозване функције удаљености).

Метод памти све податке (у даљем тексту тачке) тренинг скупа заједно са њиховим ознакама. Да би класификовао нову тачку, он ће проћи кроз читав скуп тачака, у односу на задату меру удаљености пронаћи k најближих (то су најближи суседи), а затим на основу класе којој припада највише од изабраних k суседа класификовати нов податак. Употреба геометријске удаљености (нпр. еуклидско растојање) за дефинисање растојања међу тачкама не мора увек бити најповољнија или чак могућа: врста уноса може, на пример, бити текст где на први поглед није јасно како треба мерити удаљеност. Стога метрику удаљености треба пажљиво бирати, за сваки проблем појединачно. У случају растојања међу текстуалним подацима може се користити такозвано косинусно растојање или метрике изведене из њега, али о томе неће бити речи на овом месту.

Најбољи избор k зависи од података; генерално гледано, веће вредности k смањују непрецизности и грешке, али су зато границе међу класама мање јасне. Што се тиче окружења *Orange*, параметар k може се мењати у прозору који се појављује кликом на оператор *KNN*, у пољу *Number of neighbors*, што је приказано на слици 7.1. У овом прозору могуће је подешавати и метрику која ће се користити за мерење растојања међу појединачним подацима. У овом примеру реч је о еукликдској метрици.

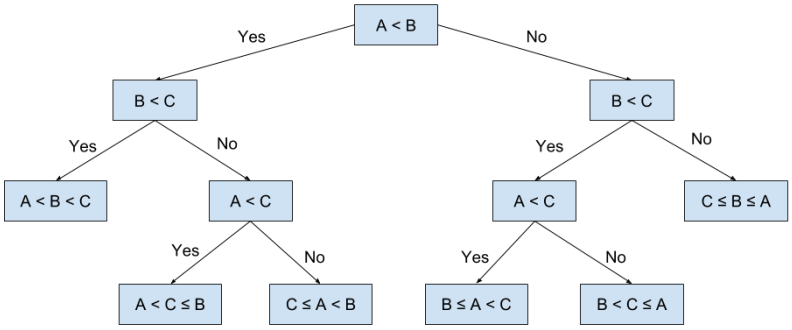


Слика 7.1: Избор параметра *k*

7.2 Стабла одлучивања

Стабла одлучивања (енг. *decision tree*) представљају модел који се веома често користи у машинском учењу. То је једна од најједноставнијих, а ипак најкориснијих модела машинског учења са веома широком применом. Генерално, стабла одлучивања се граде путем алгоритамског приступа који идентификује начине поделе скупа података на основу различитих услова. Стабла одлучивања су метод који се користи код надгледаног учења и који се примењује и за задатке класификације као и за задатке регресије. Циљ је креирати модел који предвиђа вредност циљне променљиве учењем једноставних правила одлучивања изведених из скупа за тренирање. Модел доноси одлуке на основу вредности атрибута, почев од најинформативнијих ка оним мање информативним. Велика предност овог модела огледа се у његовој интерпретабилности. Другим речима, у сваком тренутку је јасно на основу чега је модел донео неку одлуку. Ово својство интерпретабилности јако је важно код осетљивих проблема као што је рецимо медицинска дијагностика, када је лекарима потребно објашњење зашто је модел донео одређене одлуке.

Стабло одлучивања класификује сваки нов податак прослеђујући га кроз тестове од корена ка листовима стабла. У сваком чвору стабла налази се по један тест, који има два или више исхода. Сваком исходу одговара по једна грана стабла, и она води до следећег чвора. Листови су означени вредностима које представљају предвиђања стабла (ознакама класе у случају класификације или нумеричким вредностима у случају регресије). Сваки податак, у зависности од исхода, креће се од корена, кроз одговарајуће чворове, све док не стигне до листа који представља конкретно предвиђање. Илустрација стабла одлучивања за проблем бинарне класификације дата је на слици 7.2.

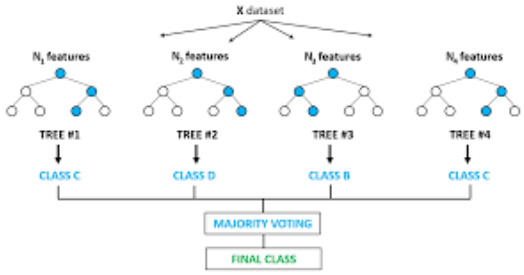


Слика 7.2: Стабло одлучивања

7.3 Метод случајних шума

Метод случајних шума (енг. *random forest*) представља метод машинског учења који се користи код проблема надгледаног учења како за класификацију тако и за регресију. Као што је шума сачињена од дрвећа тако је и метод случајних шума састављен од већег броја метода стабла одлучивања. Метод случајних шума тренира велики број стабала одлучивања на различитим, случајно изабраним, подскуповима полазног скупа података, и у време предвиђања агрегира појединачна предвиђања сваког од стабала за дати податак. Агрегирање се у случају класификације може дефинисати као избор најфреквентније класе, док се у случају регресије агрегација може дефинисати упросечавањем добијених нумеричких вредности. То је метода ансамбла (ансамбл се састоји од свих натренираних стабала одлучивања) која је боља од једног стабла одлучивања јер смањује преприлагођавање (енг. *overfitting*) обучавањем већег броја класификатора. Већи број стабала у шуми доводи до робуснијег метода машинског учења који обично има мању грешку него једноставнија стабла. Важно је напоменути да се агрегирањем већег броја стабала у великој мери губи на интерпретабилности коју она нуде.

Илустрација случајне шуме са четири стабла одлучивања дата је на слици 7.3.



Слика 7.3: Случајна шума

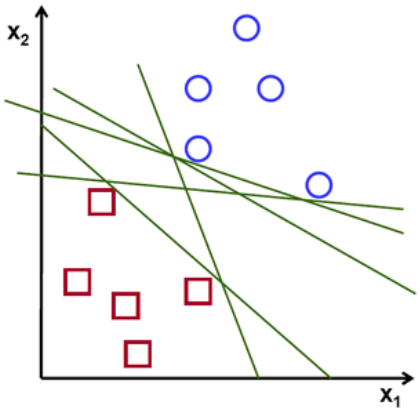
7.4 Метод потпорних вектора

Метод потпорних вектора (енг. *support vector machines – SVM*) је важан метод надгледаног учења који има за циљ да пронађе хиперраван у n-димензионалном простору (n - број атрибута, карактеристика) који јасно разврстава тачке података. Оптимална хиперраван је подједнако удаљена од најближих података сваке класе. Овај модел се може користити и за регресију и за класификацију, али се више користи у проблемима класификације. У наставку ће бити речи о најједноставнијем случају бинарне класификације.

Димензија хиперравни зависи од броја атрибута. Ако је димензија улазних података n (ово одговара броју атрибута сваки појединачни податак), хиперраван ће јасно бити димензије $n-1$; Ако су улазни подаци дводимензионални (рецимо тачке у еуклидској равни), тада је хиперраван линија; Ако су улазни подаци тродимензионални (као пример могу послужити тачке у тродимензионом простору), тада хиперраван постаје дводимензионална раван.

Потпорни вектори су подаци који су најближи хиперравни и утичу на положај и оријентацију хиперравни. Брисањем или додавањем потпорних вектора промениће се положај хиперравни. Ово су тачке које одерђују хиперраван. Интуитивно их је најлакше замислити као држаче који дају потпору и истовремено фиксирају раздвајајућу хиперраван.

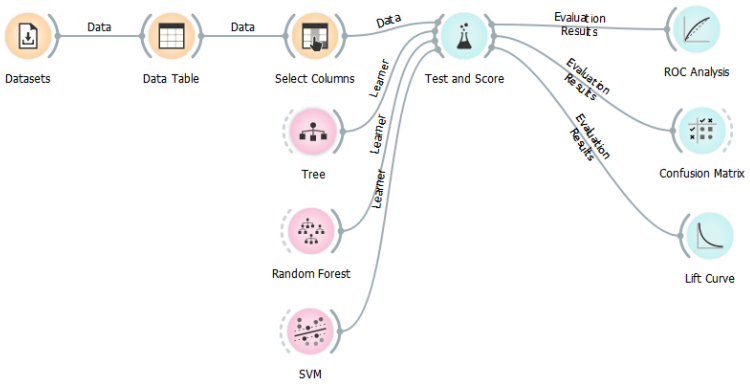
При коришћењу методе потпорних вектора је након тренинга и избора саме раздвајајуће хиперравни могуће одбацити све податке осим потпорних вектора, што позитивно утиче на временску и просторну сложеност примене самог алгоритма. Илустрација могућих раздвајајућих хиперравни у случају дводимензионог скупа података дата је на слици 7.4. Јасно, идеалана раздвајајућа хиперраван била би она која је на највећем растојању од најближе тачке података за тренинг. Тачке се класификују у зависности од тога где се налазе у одоносу на раздвајајућу хиперраван.



Слика 7.4: Хиперравни

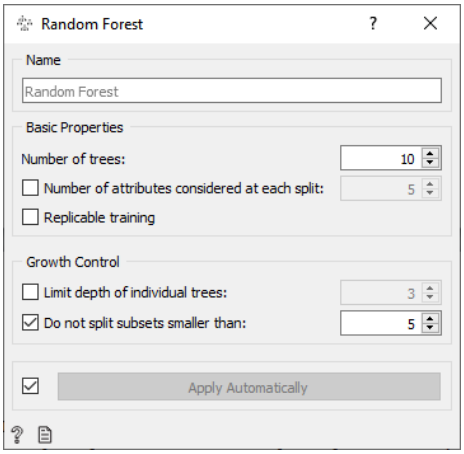
Ради илустрације, у наставку је пример примене наведена три модела на проблем регресије и проблем класификације.

Најпре је потребно посматрати проблем класификације података на скупу *Iris*. Даље, на овај скуп података треба применити све три наведене методе, уз алгоритме за евалуацију – матрица конфузије, ROC крива и лифт кива, као на слици 7.5, да би на конкретном примеру било јасно која је метода најпогоднија за решавање овог проблема.



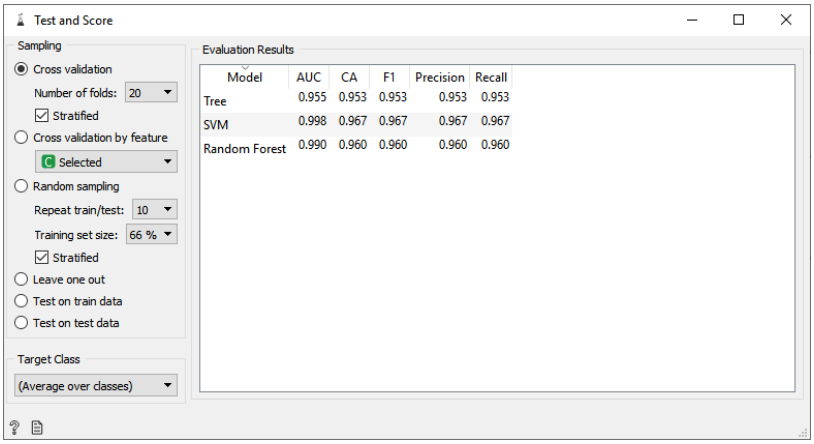
Слика 7.5

Када је реч о методу случајних шума, број стабала одлучивања може се мењати, у прозору који се отвара кликом на оператор *Random forest*, у пољу *Number of trees*, што је приказано на слици 7.6. У овом прозору могуће је подешавање и неких напреднијих параметара, о којима неће бити речи.



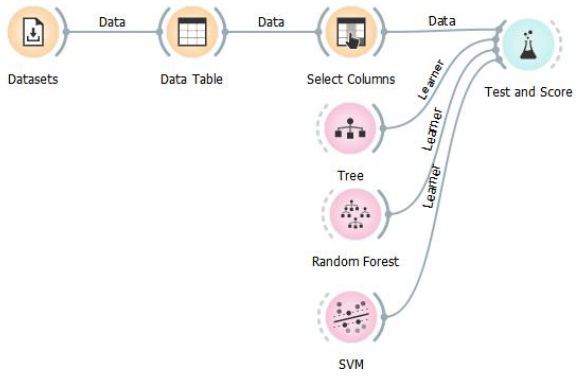
Слика 7.6: Избор броја стабала

Кликом на оператор *Test and Score* добијају се информације о оцени ова три модела. У колони *CA* (*Classification Accurancy*) види се тачност класификације, односно удео тачно класификованих података за дат скуп података, за ове три методе на основу резултата сва три модела евалуације. Дакле, прецизност методе стабло одлучивања је 0.953 (95.3%), прецизност методе потпорних вектора је 0.967 (96.7%), а прецизност методе случајних шума је 0.960 (96.0%). Одавде се закључује да за проблем класификације скупа података *Iris* најпрецизније резултате даје метода потпорних вектора, затим метода случајних шума, а најмању прецизност даје метода стабла одлучивања. Додатно, овде је видљиво и повећање прецизности од готово 1% које доноси агрегирање дрвета одлучивања у ансамбл.



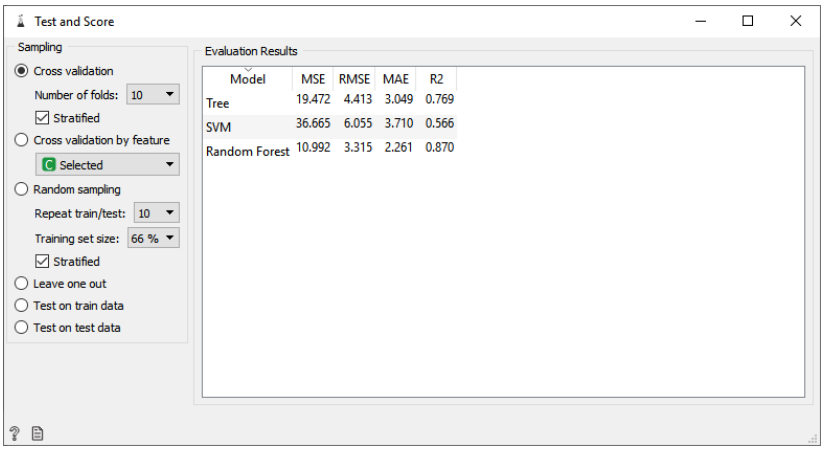
Слика 7.7: Евалуација

Исти поступак је сада потребно применити на проблем регресије скупа података *Housing* (Слика 7.8).



Слика 7.8

Кликом на оператор *Test and Score* добијају се информације о оцени ова три модела. Очигледно је да је средњеквадратна грешка најмања код метода случајних шума, што значи да је овај модел најпогоднији за решавање задатог проблема, док метода потпорних вектора даје највећу грешку при обучавању на овом скупу података. На овом месту важно је истаћи позитивне ефекте агрегирања стабала одлучивања у случајну шуму на основу повећања површине испод ROC криве.



Слика 7.9: Евалуација

7.5 Линеарна регресија

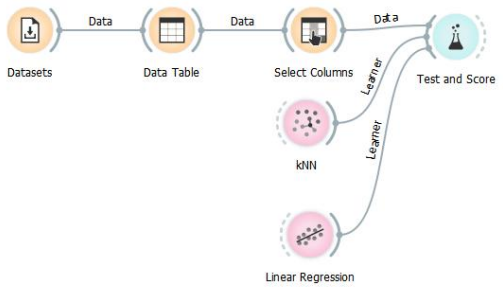
Линеарна регресија (енг. *linear reggression*) је метода надгледаног машинског учења која проналази линеарни однос који важи између улаза и излаза. Отуда име линеарна регресија. Линеарна регресија је погодна за предвиђање резултата који је непрекидне вредности, као што је предвиђање цене некретнине. Његов резултат предвиђања може бити било који реалан број.

Поставља се питање зашто линеарна регресија не може да се користи за решавање проблема класификације. Циљ линеарне регресије јесте да пронађе однос између улазних варијабли и циљне варијабле. Линеарна функција добијена из скупа за тестирање има за циљ да смањи удаљеност између предвиђене вредности и стварне вредности. Најпре, проблем прави то што је код линеарне регресије предвиђена вредност континуирана, а не пробабилистичка (вероватносна). У проблему бинарне класификације оно што је потребно предвидети је вероватноћа да ће се исход догодити. Вероватноћа се креће између 0 и 1, где је вероватноћа да се нешто сигурно догоди 1, а 0 је нешто што се скоро сигурно неће догодити. У линеарној регресији предвиђени број може бити изван 0 и 1.

Додатно, као важно својство линеарне регресије на овом месту треба истаћи то да је линеарна регресија јако осетљива на одударајуће податке. Наиме, с обзиром да линеарна регресија конструише линеарну функцију минимизирањем грешке предвиђања, како би умањила удаљеност предвиђене и стварне вредности, али и с обзиром на то да се у функцији грешке за линеарну регресију грешке квадрирају, сваки одударајући податак може направити велику разлику у предвиђајућој функцији.

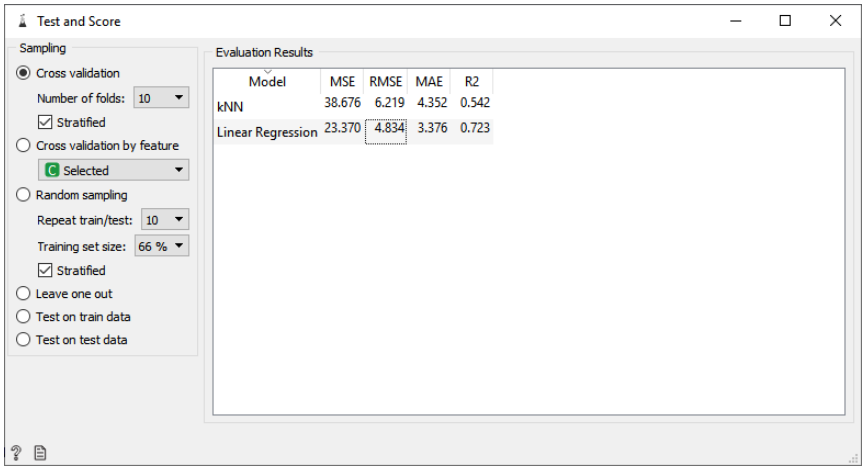
Сада је скуп података *Housing* потребно представити табеларно, као у претходном примеру, а затим ове операторе повезати са оператором *Test and Score* који се налази у одељку *Evaluate*.

Оператор *Test and Score* повезаћемо моделима k-најближих суседа (kNN) и линеарне регресије, као што је приказано на слици 7.10.



Слика 7.10

Кликом на оператор *Test and Score* могуће је видети оцене ова два модела.



Слика 7.11: Оцењивање модела

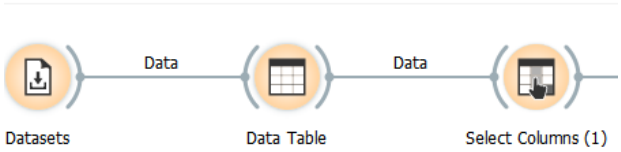
Колона *MSE* (енг. *mean square error*) је средњеквадратна грешка и заправо представља грешку средњеквадратног растојања тачака из скупа података од линеарне функције која је резултат линеарне регресије. Колона *RMSE* (енг. *root mean square error*) је корен средњеквадратне грешке, и као што је раније већ истакнуто користи се да би се анулирали квадрати који су присутни при израчунавању средњеквадратне грешке, и да би се ред величине грешке мерио на истој скали на којој се мери и циљна променљива. Ако је средњеквадратна грешка мала, то значи да добијена линеарна функција добро апроксимира скуп података.

У овом случају, средњеквадратна грешка је мања код линеарне регресије, што значи да је овај модел повољнији за коришћење на скупу података *Housing*.

7.6 Логистичка регресија

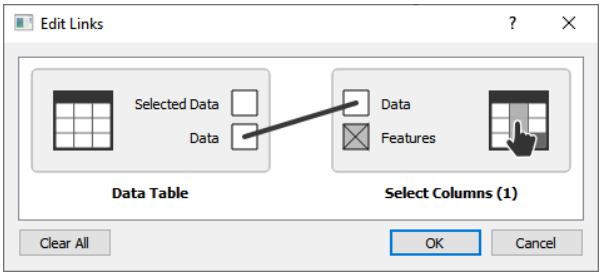
Други популарни класификатор у машинском учењу је **логистичка регресија** (енг. *logistic regression*). Логистичка регресија спада у методе класификације упркос свом називу. Ово је статистичка метода за анализу скупа података у којем постоји једна или више независних варијабли које одређују исход. Исход се мери дихотомном променљивом (у којој су само два могућа исхода).

Ради илустрације, већ поменути скуп података *Iris* је најпре потребно представити табеларно, а затим из наведеног скупа одабрати одговарајуће колоне на следећи начин:



Слика 7.12: Приказивање скупа података

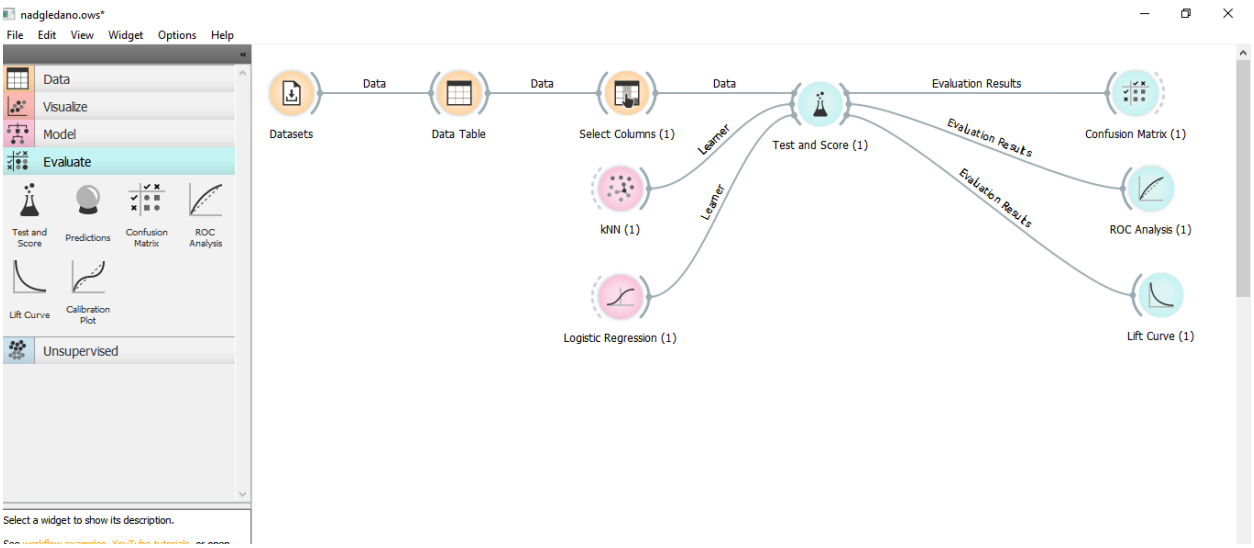
При повезивању оператора *Data Table* и *Select Columns* потребно је правилно повезати улазне и излазне податке:



Слика 7.13: Повезивање података

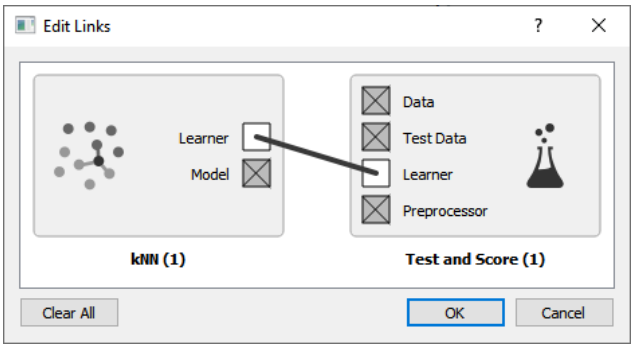
Даље је потребно повезати ове операторе са оператором *Test and Score* који се налази у одељку *Evaluate*. Овај оператор ће бити потребан ради евалуације и поређења модела који ће бити коришћемо на скупу података.

Оператор *Test and Score* је потребно повезати са једне стране са моделима *k*-најближих суседа (*kNN*) и логистичке регресије, а са друге стране операторима за евалуацију – матрицом конфузије, ROC кривом и лифт кривом, да би се, коришћењем различитих начина оцењивања извршила процена који је модел бољи за задати скуп података, као што је приказано на слици 7.14.



Слика 7.14

При повезивању оператора *Test and Score* са моделима за класификацију, опет је потребно водити рачуна о правилном повезивању података. Та веза мора бити одређена на следећи начин:

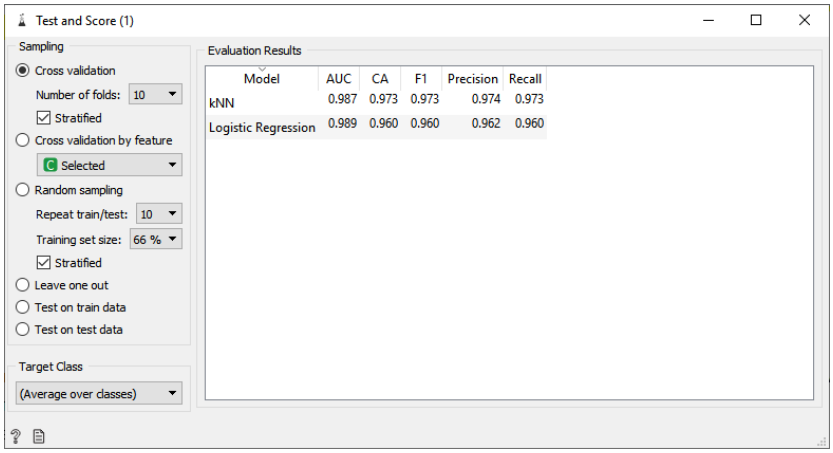


Слика 7.15: Повезивање података

Приказани прозор се појављује након клика на везу између жељених оператора.

Дакле, скуп података је повезан са моделима класификације *k*-најближих суседа и логистичком регресијом, и овим моделима је додељена команда да уче из података. Користећи методе оцењивања, могуће је упоредити који је модел повољнији за учење на овом скупу података.

Подаци о евалуацији се добијају двоструким кликом на оператор *Test and Score*:



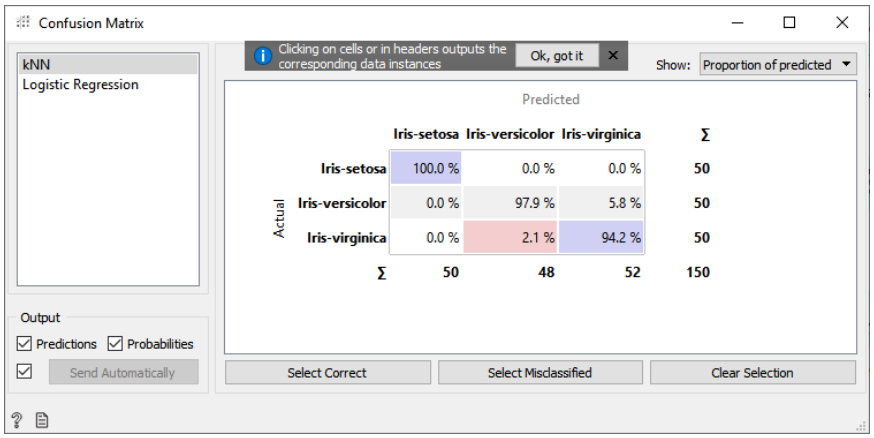
Слика 7.16: Оцењивање модела

На слици је приказана оцена ова два модела. У колони *CA* (енг. *classification accurancy*) можемо видети тачност класификације, односно удео тачно класификованих података за дати скуп података, за ове две методе на основу резултата сва три модела евалуације. Дакле, прецизност методе *k*-најближих суседа је 0.973 (97.3%), а прецизност методе логистичке регресије је 0.960 (96.0%). Додатно, метода најближих суседа на креираном скупу података поседује како већу прецизност, тако и већи одзив. Одавде се може закључити да је за класификацију на скупу података Iris погодније користити методу к-најближих суседа.

Колона *AUC* (енг. *area under receiver operating characteristic*) је заправо површина испод *ROC* криве, о којој је више речи било у претходном поглављу, док је за остале метрике јасно на шта се односе.

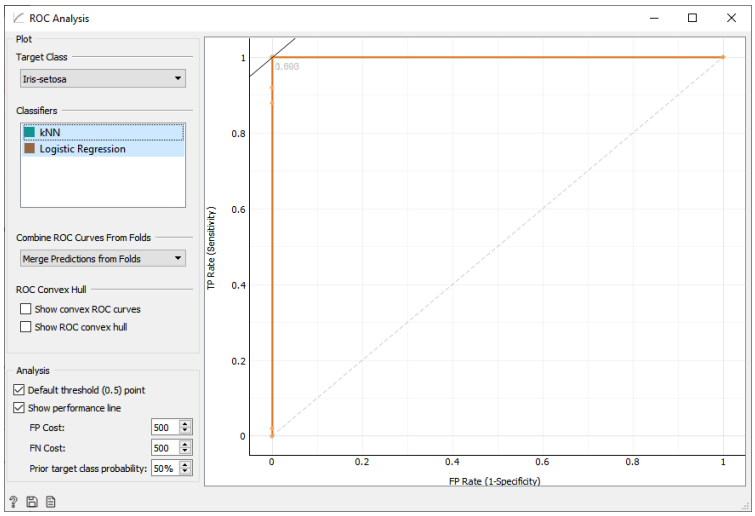
На овом примеру такође је могуће видети како изгледају посматране мере. Потребно је само кликнути на оператор који представља конкретан модел:

- Матрица конфузије:



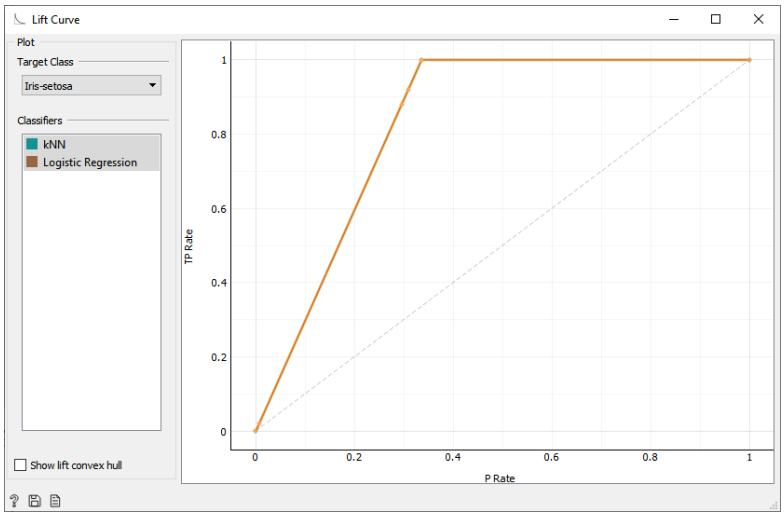
Слика 7.17: Матрица конфузије

- ROC крива:



Слика 7.18: ROC крива

- Лифт крива:



Слика 7.19: Лифт крива

7.7 Неуронске мреже

Неуронске мреже (енг. *neural networks*) су тренутно најпопуларнија метода машинског учења, применљива како на проблеме регресије тако и на проблеме класификације. При томе, чест је случај да се појам машинског учења потпуно изједначава са појмом неуронских мержа, што је свакако погрешно. Неуронске мреже се могу применити на широк спектар проблема укључујући слике, видео снимке, датотеке, базе података и још много тога. Због генерализованог приступа решавању проблема које нуде неуронске мреже, практично не постоји ограничење у областима на које се ова техника може применити. Неке од конкретних примена неуронских мрежа данас укључују препознавање слике, предвиђање путање возила, препознавање лица, препознавање рукописа, филтрирање непожељне поште, медицинска дијагностика и слично.

Неуронска мрежа је систем учења који симулира неуронске везе присутне код људи и других сисара. Концепт неуронске мреже био је инспирисан људском биологијом и начином на који неурони људског мозга функционишу и међусобно интерагују како би разумели опажања из људских чула и произвели одговарајуће акције.

Метод неуронске мреже учи из обраде многих означених примера (тј. података са ознакама, "одговорима") који се обрађују током фазе тренинга и помоћу којих закључује које су карактеристике уноса потребне више а које мање важне за предвиђање исправног резултата (било класе или нумеричке вредности, зависно од проблема који се решава). Након обраде довољног броја примера, неуронска мрежа може почети да обрађује нове улазе и успешно враћати тачне резултате. Што више примера и различитих улаза програм види, резултати ће бити тачнији. Другим речима, неуронске мреже обично захтевају нешто већу количину података за тренирање, али су стога и способније за проналажење финијих зависности у подацима који нису видљиви једноставнијим методама машинског учења.

Ради појашшњавања наведеног, нека је, на пример, потребно конструисати метод који ће препознати да ли се на слици налази аутомобил. Иако је човеку једноставно да дође до одговора на ово питање, обучити рачунар да препознаје аутомобил на слици је веома сложен процес. Дефинисати прецизан алгоритам који би овако нешто одлучивао делује као немогућ процес. Отежавајућа околност је то што постоји моного могућности како аутомобили могу изгледати на слици (различите боје, углови камере, типови аутомобила и сл.). Управо овакви примери у којима је интуитивно јасно како решити проблем, при чему човек то са лакоћом и ради, а где је јако тешко дефинисати тачан метод јасан су сигнал да је погодно употребити неуронску мрежу као модел учења.

Помоћу машинског учења, тачније неуронских мрежа, програм може научити комплексне зависности које важе на одређеном скупу података. Користећи неколико слојева неурона рашчланити слику у тачке, векторе као и компликованије обрасце - податке које рачунар може користити. Неуронска мрежа идентификује трендове који постоје у многим примерима које обрађује и класификује слике према њиховим сличностима.

Након обраде многих примера слика аутомобила током процеса тренинга, метод је обучио модел које елементе (конкретно пикселе) и њихове односе на слици је важно узети у обзир при одлучивању да ли је аутомобил присутан на слици или не. Приликом покретања модела на новој слици, неуронска мрежа користи научене везе као и јачине веза између неурона своје архитектуре (такозване тежине, реални бројеви). Слојеви неурона између улаза и излаза су оно што чини неуронску мрежу. Процес тренинга своди се на поналажење јачина свих постојећих веза унутар неуронске мреже, такозваних тежина које су реални бројеви, тако да неуронска мрежа добро обавља жељени посао.

Раније верзије неуронских мрежа биле су релативно мале дубине, састављене од једног улазног и једног излазног слоја, са обично једним скривеним слојем између. У новије време, захваљујући напредним техникама обучавања односно тренинга неуронских мрежа, могуће је натренирати знатно изражајније и дубље архитектуре, које су пак способне да откривају финије зависности међу подацима.

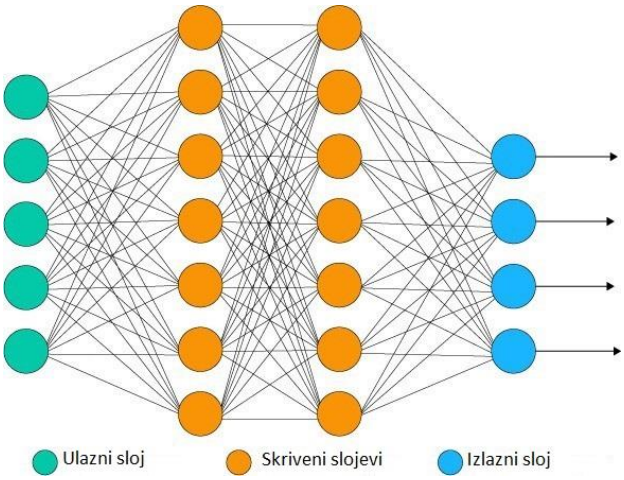
У неуронским мрежама сваки слој тренира се над различитим артибутима који су добијени као излази претходног слоја мреже. Што се дубље напредује кроз неуронску мрежу, то су сложеније карактеристике које чворови могу препознати, јер они користе и рекомбинују карактеристике из претходног слоја. Ово чини неуронске мреже учења способним да обрађују веома велике, вишедимензионалне скупове података (користећи моделе са милионима параметара).

Један од проблема које неуронске мреже најбоље решавају је обрада сирових, неозначених података, уочавање сличности и аномалија у подацима које ниједан човек није организовао у релацијској бази података нити им је икад навео име.

Неуронске мреже обично аутоматски обрађују податке у њиховом изворном облику, без људске интервенције, препроцесирања и модификација, за разлику од већине традиционалних метода машинског учења. Способност овог метода да обрађује и учи из огромних количина сирових података даје му јасну предност у односу на претходне алгоритме.

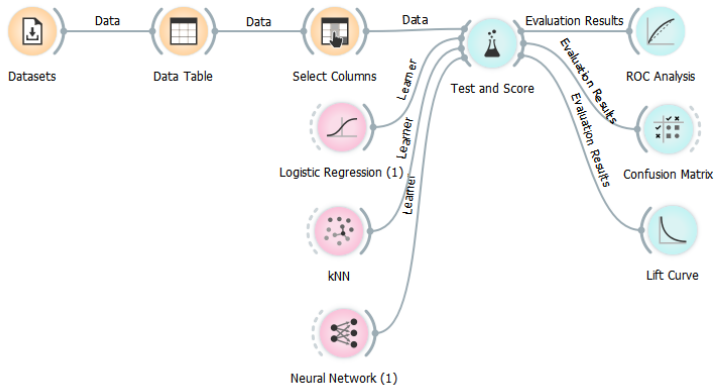
Илустрација неуронске мреже са два скривена слоја (истакнути наранџастом бојом) дата је на слици 7.20. При томе се ова терминологија, као и комплетна прича у овом раду односи на неуронске мреже у њиховом најосновнијем облику – **потпуно повезане неуронске мреже** (енг. *feed forward neural networks*) док се неуронске мреже у данашњим апликацијама могу срести у разним облицима као што су **конволутивне неуронске мреже** (енг. *convolutional neural networks*), затим **рекурентне неуронске мреже** (енг. *reccurent neural networks*), **графовске неуронске мреже** (енг. *graph neural networks*) и друге.

У наставку ће кроз окружење *Orange* бити илустрован рад са потпуно повезаним неуронским мрежама.



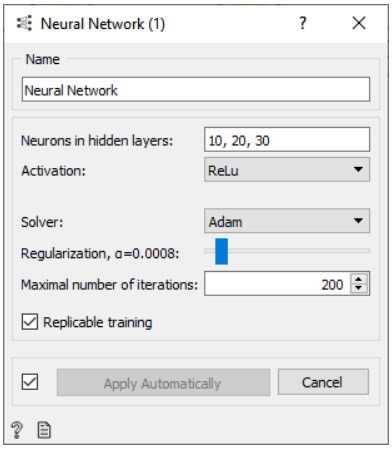
Слика 7.20: Дубока неуронска мрежа

Овде је потребно вратити се на први пример из претходне главе (пример 1 из главе 7) – проблем класификације скупа података *Iris*. На начин који је објашњен у претходној глави, потребно је додати и модел неуронске мреже у овај пример, као што је приказано на слици 7.21.



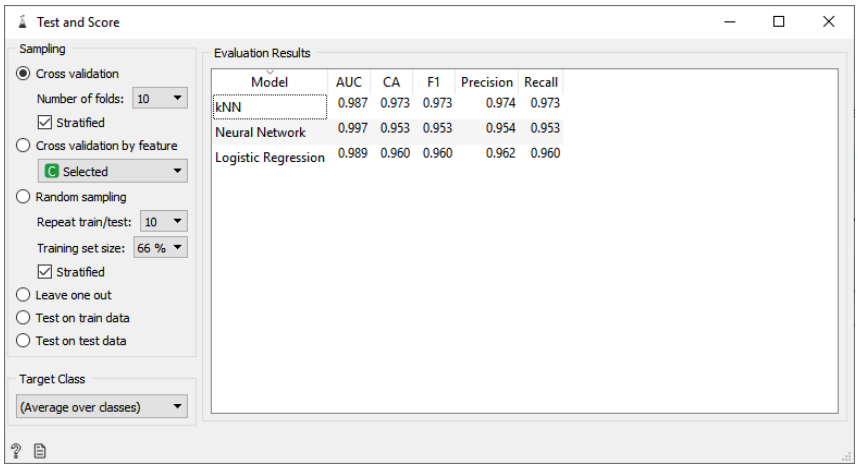
Слика 7.21

У окружењу *Orange* прозор у коме је могуће подешавати параметре неуронске мреже отвара се кликом на оператор *Neural Network* (слика 7.22). Број скривених слојева, конкретније број неурона у сваком од скривених слојева подешава се у пољу *Neurons in hidden layers.* У овом конкретном случају реч је о 3 скривена слоја, са по 10, 20 и 30 неурона, редом. У наведеном прозору могуће је подешавати и неке додатне, напредније параметре, о којима неће бити реч.



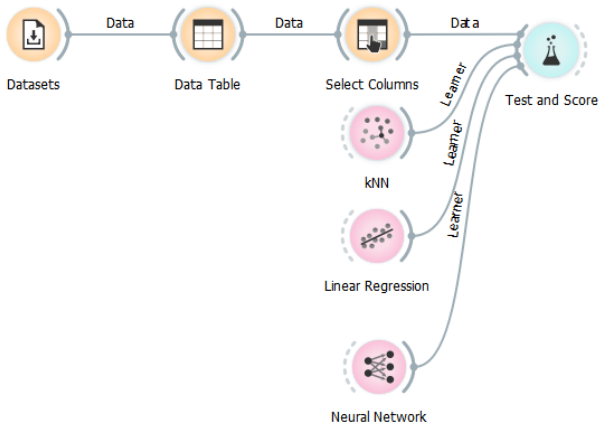
Слика 7.22: Параметри неуронске мреже

Кликом на оператор *Test and Score* добијају се информације о оцени ова три модела. У колони *CA* (*Classification Accurancy*) може се видети тачност класификације, односно удео тачно класификованих података за дат скуп података, за ове три методе на основу резултата сва три модела евалуације. Дакле, прецизност методе *k*-најближих суседа је 0.973 (97.3%), прецизност алгоритма неуронских мрежа је 0.953 (95.3%), а прецизност методе логистичке регресије је 0.960 (96.0%). Одавде се закључује да за проблем класификације скупа података *Iris* најпрецизније резултате даје метода *k*-најближих суседа, затим метод неуронских мрежа, а најмању прецизност даје метода логистичке регресије.



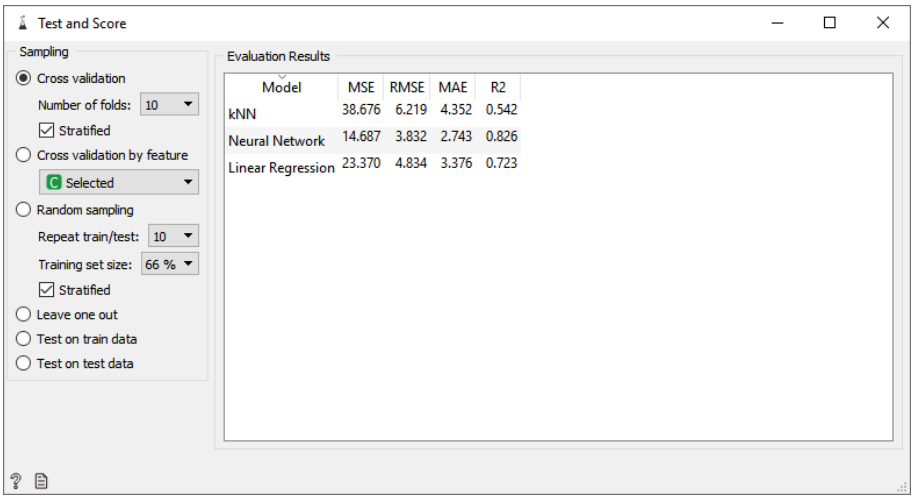
Слика 7.23: Оцењивање модела

Даље је потребно исти поступак поновити за други пример из претходног поглавља – проблем регресије на скупу података *Housing*. Дакле, и овде је потребно додати модел неуронских мрежа, ради упоређивања успешности овог модела у односу на методе k-најближих суседа и линеарне регресије, које су имплементиране раније, а као што је приказано на слици 7.24:



Слика 7.24

Кликом на оператор *Test and Score* добијају се информације о оцени ова три модела. Очигледно је да је средњеквадратна грешка најмања код неуронских мрежа, што значи да је овај метод најпогоднији за решавање задатог проблема, од ова три наведена метода. Додатно, и остале метрике у овом примеру показују доминантност неуронских мрежа. Значајно је истаћи велику разлику у коефицијенту детерминације у односу на раније дефинисане моделе ($R2$ скор).



Слика 7.25: Оцењивање модела

7.8 Алгоритам К-средина

Кластеровање је врста ненадгледаног учења. Подсећања ради, метода ненадгледаног учења је метода у којој се скупови података састоје од улазних података без ознака (припадајуће класе у случају класификације или нумеричке вредности у случају регресије). Обично се користи као процес проналажења смислене структуре односно груписања података према међусобним сличностима.

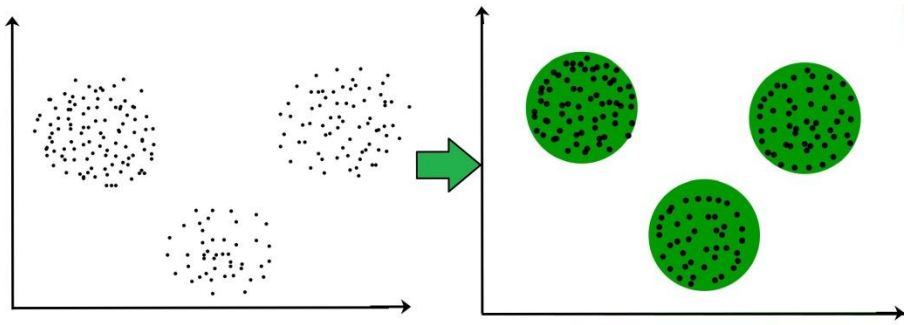
Кластеровање врши поделу података у више група тако да су подаци из једне групе слични другим подацима у истој групи а различите од података у другим групама. То је у основи груписање неозначених података на основу сличности и различитости међу њима. За један скуп података може постојати више начина за груписање односно кластеровање података, зависно од тога који критеријум се користи за одређивање сличности, односно начина мерења растојања између њих.

Алгоритми ненадгледаног учења овог типа имају широк спектар примена и прилично су корисни за решавање проблема у стварном свету као што су откривање аномалија, груписање докумената или проналажење купаца са заједничким интересима на основу њихових претходних куповина.

Неки од најчешћих метода кластеровања су:

- алгоритам К-средина (енг. *K means clustering*);
- хијерархијско кластеровање (енг. *hierarchical clustering*).

Илустрација процеса кластеровања са три, јасно одвојена кластера дата је на слици 7.26.



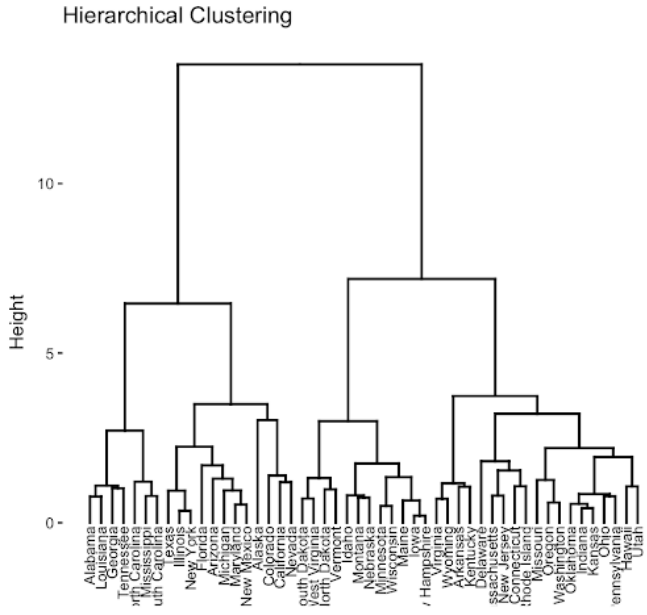
Слика 7.26: Кластеровање података

Алгоритам К-средина (енг. *K-means algorithm*) представља јако интуитиван метод кластеровања који је изузетно једноставан за имплементацију а при томе врло ефикасан. То су главни разлози који објашњавају одакле овом алгоритму толико популарности. Метод обично почиње случајним распоређивањем тачака у кластере, а касније се у сваком кораку свака тачка додељује најближој центроиди, где су центроиде тачке које интуитивно представљају центре кластера и дефинисане су као просеци свих припадајућих тачака тог кластера. Овај метод се среће у скуповима података који могу бити представљени као тачке у реалном n-димензионом простору, где се као метрика растојања обично користи еуклидско растојање. Код ове методе, избор броја кластера (број k) често може представљати проблем. Постоје неке хеуристике избора параметра k, међутим о њима неће бити речи на овом месту.

7.9 Хијерархијско кластеровање

Хијерархијско кластеровање (енг. *hierarchical clustering*) групише податке у стабло кластера. Главна предност хијерархијског кластеровања је та што овај модел у великој мери олакшава избор броја кластера. Поред тога, омогућава цртање дендограма. Дендограми су визуализације бинарног хијерархијског груписања.

Хијерархијско кластеровање започиње тако што се свака тачка из скупа података третира као засебан кластер. Затим, се идентификују два најсличнија кластера која које спаја у један кластер. Ови кораци се настављају док се сви кластери не споје. Циљ је створити хијерархијски низ угњеждених кластера. Након креираног дендограма доста је лакше определити се за број кластера у које треба поделити податке (на основу визуелизованих растојања између њих).

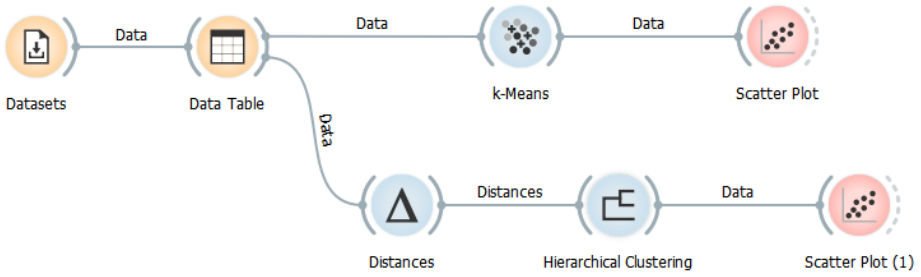


Слика 7.27: Хијерархијско кластеровање

У наставку је на примеру конкретног скупа података *Iris* илустровано како ово функционише у пракси. Користећи споменута два алгоритма, потребно је поделити овај скуп података у групе – кластере.

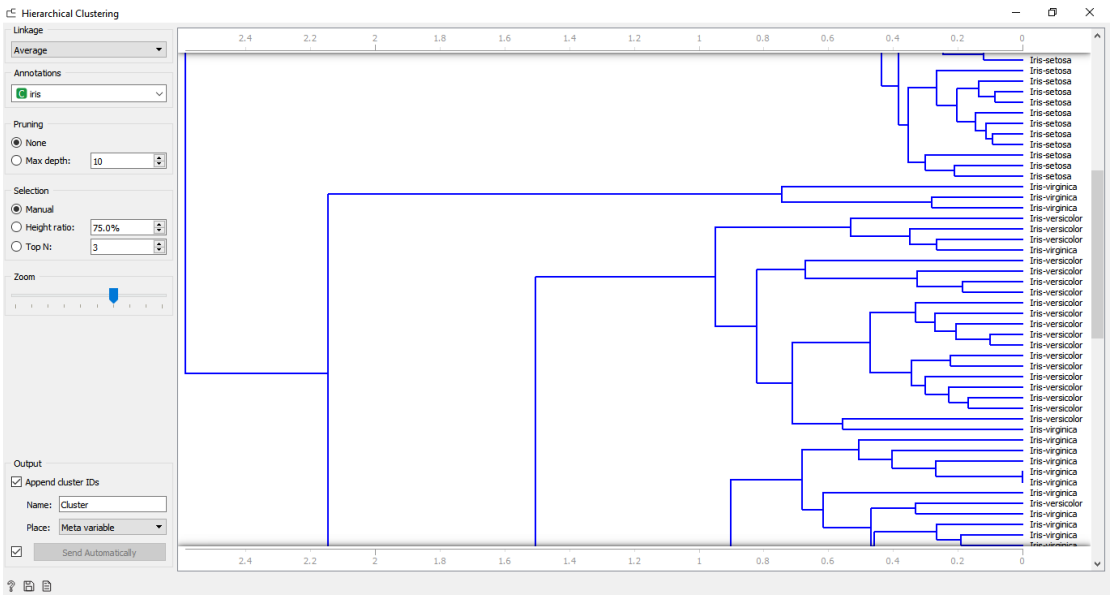
Најпре је потребно учитати скуп података, а затим га приказати табеларно, ради прегледности. Даље, потребно је одабрати оператор *k-Means* из менија који се налази на левој страни прозора, под називом *Unsupervised*, а затим и оператор *Scatter Plot* за визуелизацију овог модела.

На сличан начин потребно је одабрати и оператор *Hiererchical Clustering* који се такође налази у менију под називом *Unsupervised*. Пре него што се овај оператор изабере, потребно је укључити и оператор *Distances* који се користи за мерење растојања или дужина, зависно од проблема. И овај модел ће бити визуелизован помоћу оператора *Scatter Plot* (Слика 7.28).



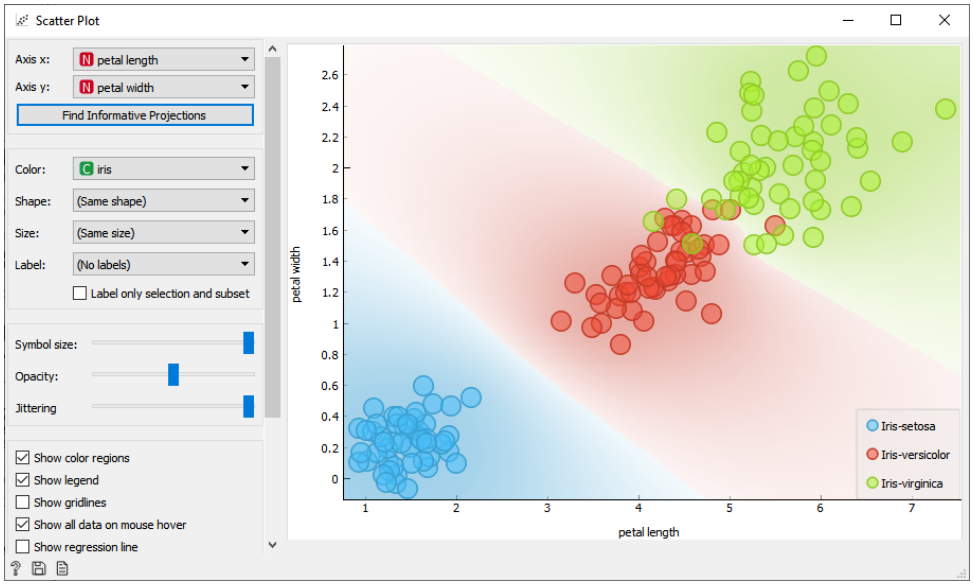
Слика 7.28

Кликом на оператор *Hiererchical Clustering* приказује се прозор на коме се налази већ споменути дендограм, односно дијаграм на ком су графички представљени кластери односно групе овог скупа података. На y оси дендограма налазе се појединачни подаци (конкретно цветови) док се припадност кластерима тих података јасно очитава са x осе. Илустрација наведеног дата је на слици 7.29.

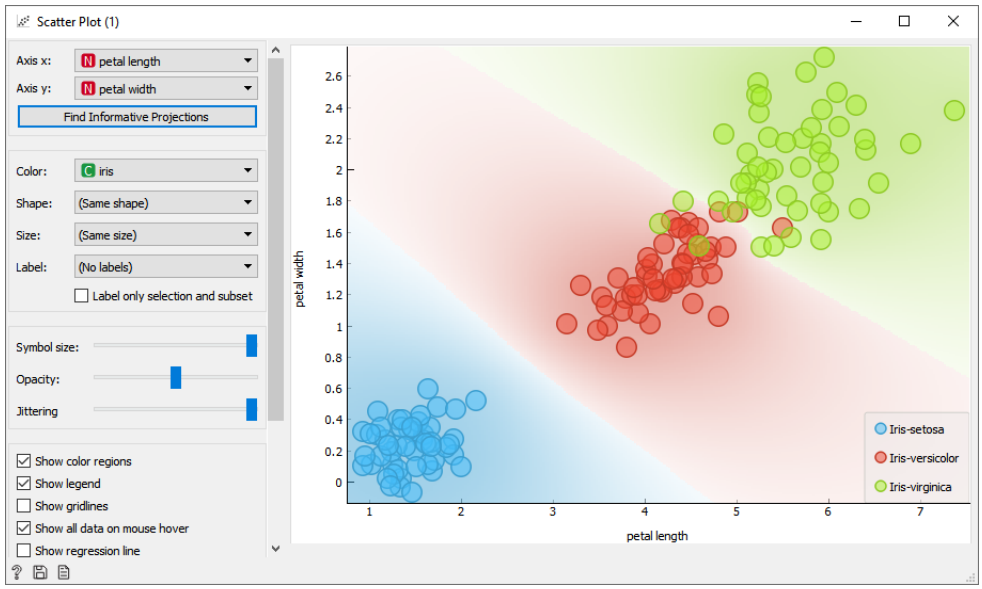


Слика 7.29: Дендограм

Даље је могуће упоредити графике добијене коришћењем ова два модела кластеровања. С обзиром на димензионалност коришћеног скупа података, сасвим је довољно ограничити се за приказ података у две димензије избором два конкретна атрибута. Конкретно, у овом примеру то ће бити атрибути *petal length* и *petal width* односно дужина и ширина латица. У том координатном систему, визуализација кластера у случају алгоритма *K* средина дата је на слици 7.30, док је визуелизација придружених кластера у случају хијерархијског кластеровања дата на слици 7.31. Овде треба напоменути да је број кластера постављен на 3, с обзиром на то да скуп података који се користи садржи податке о три различите групе цветова.



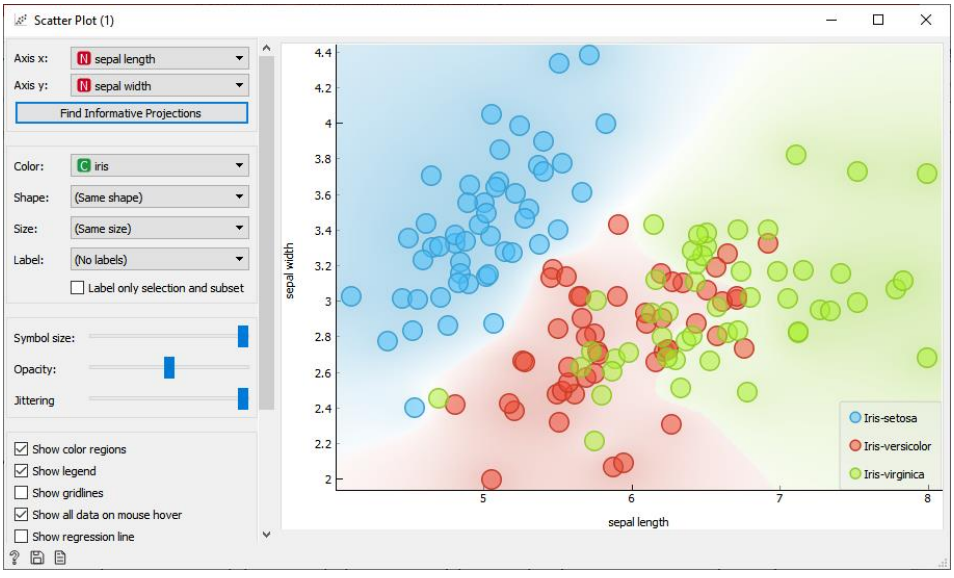
Слика 7.30: График алгоритма к-средина



Слика 7.31: График хијерархијског кластеровања

У овом случају оба алгоритма кластеровања су дала скоро па потпуно исти график. То не мора увек бити случај, поготово са комплексинијим структурама података.

Такође, у горњем левом углу овог прозора постоји опција да се одабере који атрибут ће бити приказан на којој оси. Варирањем понуђених атрибута могуће је доћи до закључка у односу на које карактеристике се подаци из скупа за тестирање могу најбоље поделити. Приказ додељених кластера алгоритма *K* средина у координатном систему дужине и ширине чашичних листића дат је на слици 7.32.



Слика 7.32: График са другим атрибутима на осама

Глава 8: Закључак

Машинско учење као област у експанзији, има значајан утицај на развој многих других области, и то не само оних уско повезаних са рачунарством. Велики број компанија примењује методе машинског учења у својим апликацијама и програмима. Све већа примена машинског учења у индустрији уско је повезана са све интензивнијим академским истраживањима ове области, што резултује значајним бројем публикација и радова у академским круговима. Машинско учење већ сада има велики утицај на области са којима на први поглед нема додирних тачака, као што су медицина или аутомобилска индустрија. Променом приступа решавању већ постојећих проблема, уз коришћење машинског учења, нађена су нова, квалитетнија решења, а примена машинског учења обезбедила је неке нове функционалности које су до сада биле готово незамисливе. Ово најбоље илуструје улога машинског учења у функционисању аутономних возила, функционалност која би без примене машинског учења била на нивоу научне фантастике. Такође, машинско учење има огроман утицај на развој медицине, доминантно кроз област биоинформатике. Коришћењем техника машинског учења померене су границе у решавању многих високодимензионих проблема, доминантних у биоинформатици.

На основу свега прехтодно наведеног може се извести закључак да ће машинско учење представљати основ за даљи резвој не само вештачке интелигенције већ и рачунарства уопште. Проблеми који тренутно постоје у овој области односе се пре свега на то што је она углавном заступљена у ужем кругу људи који се баве овом облашћу, комерцијално или академски. Такође, материјали за учење нису широко доступни, барем не у одговарајућем облику, јер се јавља проблем налажења једноставних материјала за кориснике који нису уско стручни. Систематско излагање тема из области машинског учења у овом раду може помоћи при савладавању основних концепата ове области. Све теме изложене у раду су јавно доступне и у облику електронских лекција чији је циљ лакше савладавање основа машинског учења, и то су тренутно једини доступни материјали овог обима и формата на нашем језику. Електронске лекције налазе се на адреси: http://edusoft.matf.bg.ac.rs/eskola_veba/#/course-details/ml . Приближавање машинског учења већем броју нових корисника требало би да учврсти лидерску позицију ове области у рачунарству и да допринесе њеном даљем развоју.

Литература:

[1] Јаничић, Предраг, Николић, Младен, Вештачка интелигенција, Београд 2020.

[2] Николић, Младен, Зечевић, Анђелка, Машинско учење, Београд 2019.

[3] Goodfellow, Ian и др. Deep Learning, MIT Press 2016.

[4] Лабораторија за биоинформатику, Факултет рачунарских и информационих наука, Универзитет у Љубљани, https://orange.biolab.si/