**Увод**

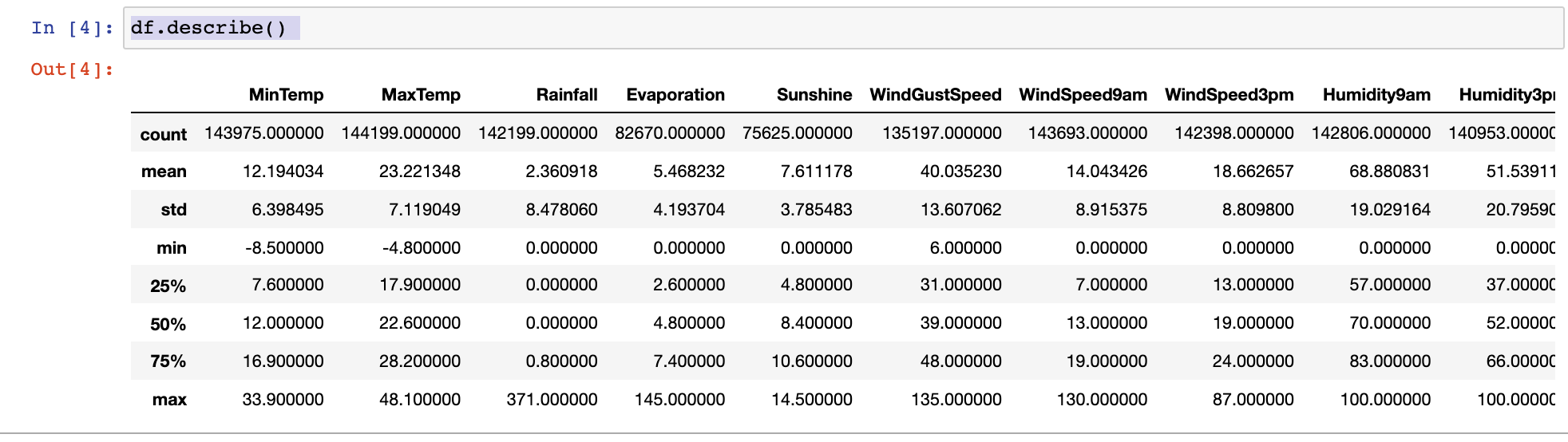
Падавине имају значајан утицај на друштво. Фестивали, различите активности и спортске утакмице примери су догађаја на које падавине веома утичу. Стога се ова тема интензивно проучава. Циљ ове анализе је да се предвиди да ли ће следећег дана падати киша, на основу временских запажања током претходног дана. Јасно се види да се проблем може посматрати као проблем бинарне класификације (1 = Киша и 0 = Без кише). Сам скуп података садржи информације о времену прикупљене током 10 година са бројних метеоролошких станица у Аустралији.

Скуп података чине 23 колоне приказане у следећој табели:

|  |  |  |
| --- | --- | --- |
| **Име** | **Значење** | **Јединица** |
| Датум | Датум посматрања | N/A |
| Локација | Локација метеоролошке станице | N/A |
| MinTemp | Минимална температура од 24 сата до 9 сати. Понекад познато само до најближег степена | °C |
| MaxTemp | Максимална температура од 24 сата до 9 сати. Понекад познат само до најближег степена | °C |
| Rainfall | Падавине у 24 сата до 9 сати. Понекад познат само до најближег милиметра | *mm* |
| Evaporation | Испаривање "класе А" у 24 сата до 9 сати ујутру | *mm* |
| Sunshine | Јарко сунце током 24 сата до поноћи |  |
| WindGustDir | Правац најјачег удара ветра у 24 сата до поноћи | 16 тачки компаса |
| WindGustSpeed | Брзина најјачег налета ветра у 24 сата до поноћи |  |
| WindDir9am | Смер ветра у 9 сати | 16 тачки компаса |
| WindDir3pm | Правац ветра у 15 часова | 16 тачки компаса |
| WindSpeed9am | Брзина ветра у 9 сати |  |
| WindSpeed3pm | Брзина ветра у 15 часова |  |
| Humidity9am | Релативна влажност у 9 сати | % |
| Humidity3pm | Релативна влажност у 15 сати | % |
| Pressure9am | Атмосферски притисак смањен је на средњи ниво мора у 9 сати | hPa |
| Pressure3pm | Атмосферски притисак смањен је на средњи ниво мора у 15:00 | hPa |
| Cloud9am | Део неба заклоњен облаком у 9 сати | Okta |
| Cloud3pm | Део неба заклоњен облаком у 15 часова | Okta |
| Temp9am | Температура у 9 часова | °C |
| Temp3pm | Температура у 3 сата ујутру | °C |
| RainToday | Да ли је у току данашњег дана у 24 сата до 9 сати падало више од 1 мм падавина | 0 = Не, 1 = Да |
| RainTomorrow | Да ли је следећег дана падавина прешла 1 мм у периоду од 24 сата до 9 сати ујутру | 0 = Не, 1 = Да |

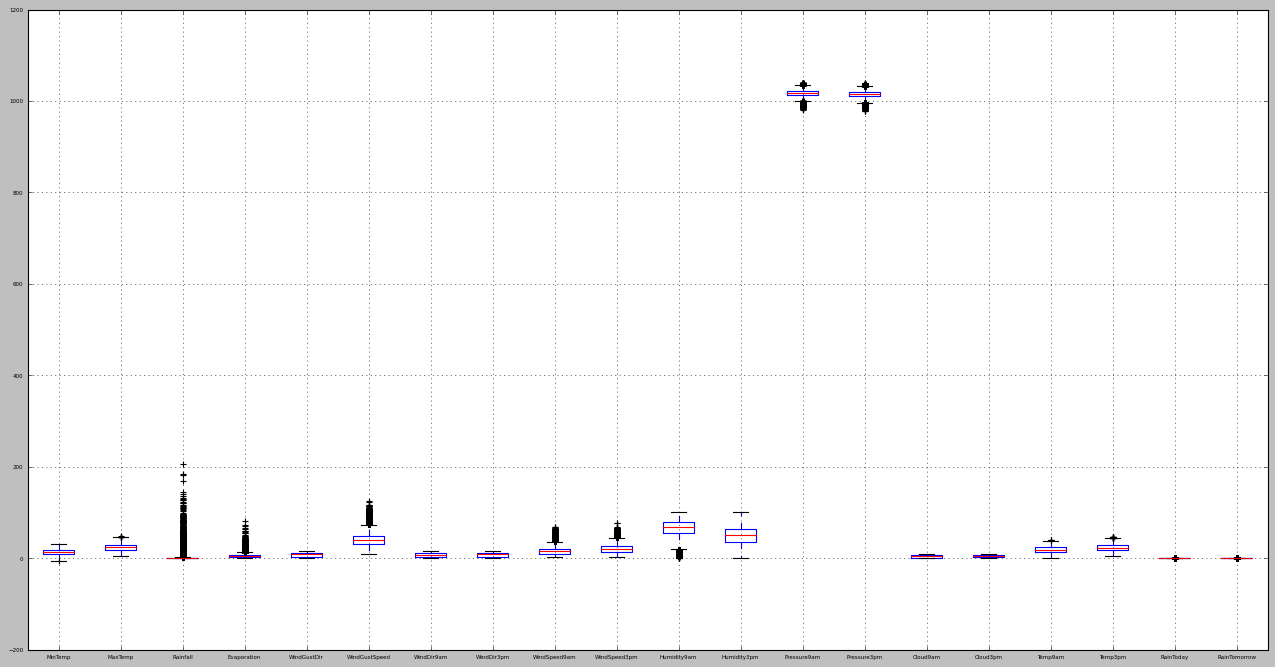
**Истраживање скупа података**

Коришћењем метода *describe()* приказују се значајне статистике као што су перцентиле, стандардно одступање, средња вредност…



*Значајне статистике*

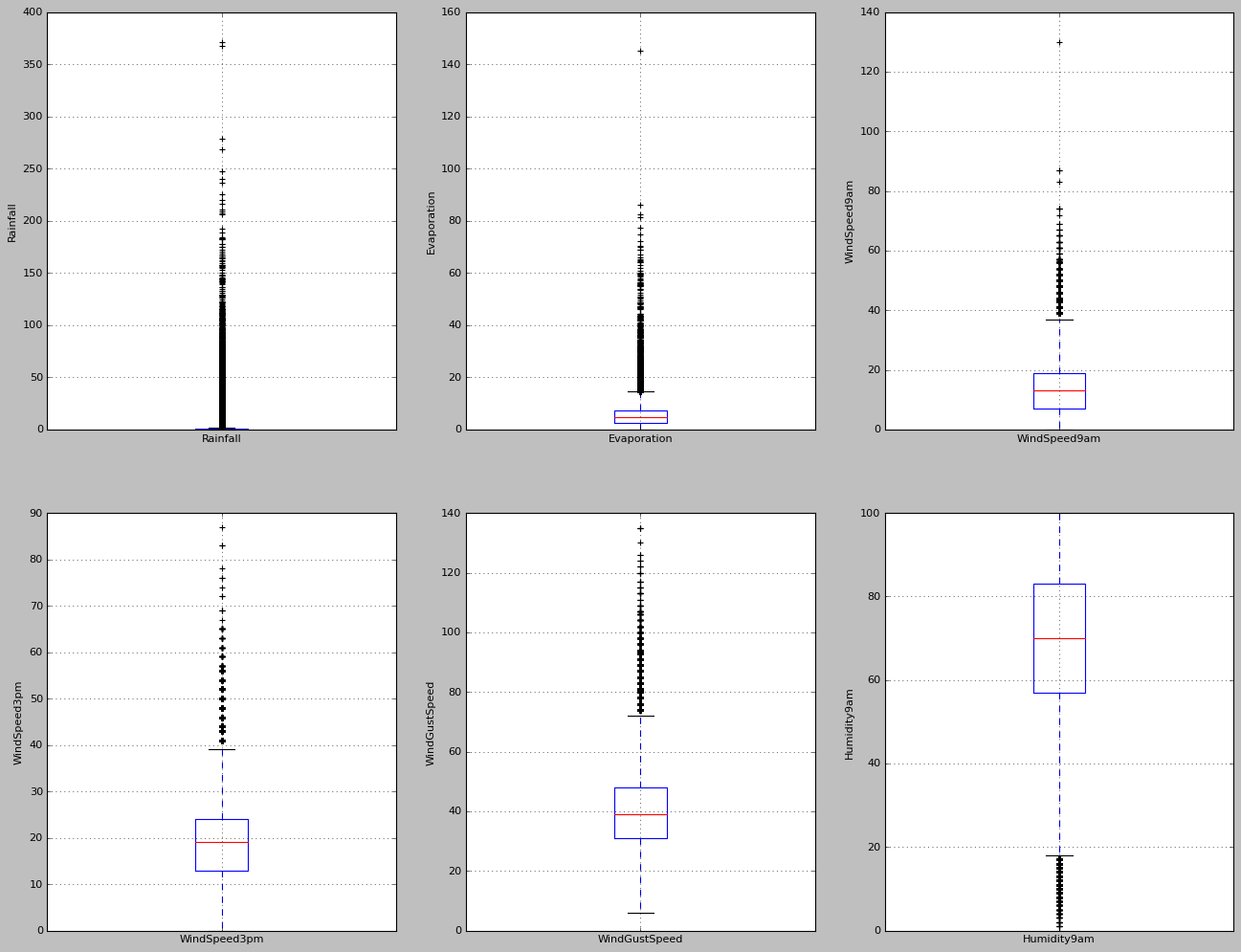
На основу приложених резултата може се закључити да више атрибута садржи елементе ван границе (*енгл. Outliers*). Једноставним позивом метода *boxplot()* могу се исцртати кућице (*енгл*. *Boxplot*) свих атрибута, како би се лакше детектовали елементи ван граница.



*Кутијасти дијаграми свих атрибута*

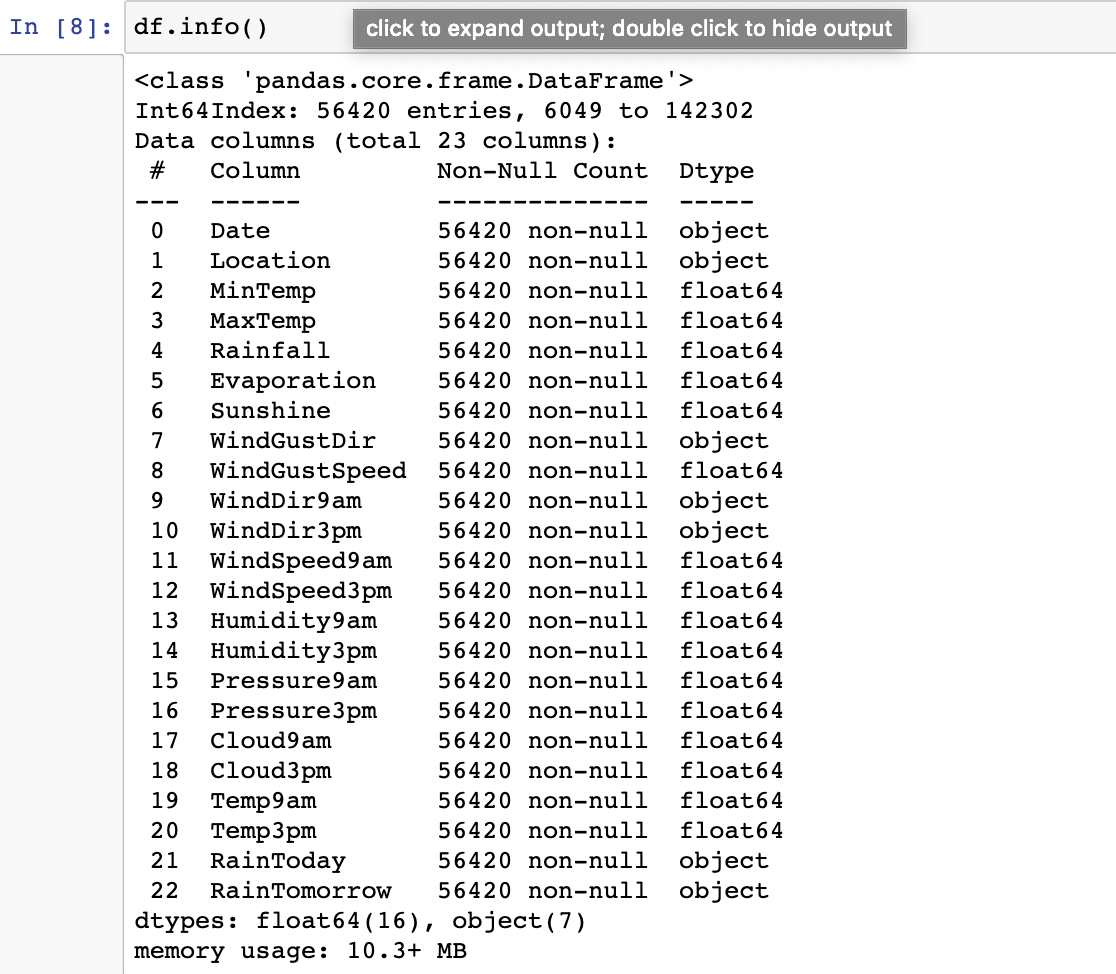
На основу анализе статистика и кутијастих дијаграма може се закључити да атрибути: Rainfall, Evaporation, WindSpeed9am, WindSpeed3pm, Humidity9am и WindGustSpeed поседују елементе ван граница.

Ради детаљнијег прегледа издвојени су поменути атрибути.



*Кутијасти дијаграми*

Функција *info()* се користи за штампање сажетка *DataFrame-a*. Овај метод штампа информације о типу података, вредностима које нису нула и о искоришћењу меморије.

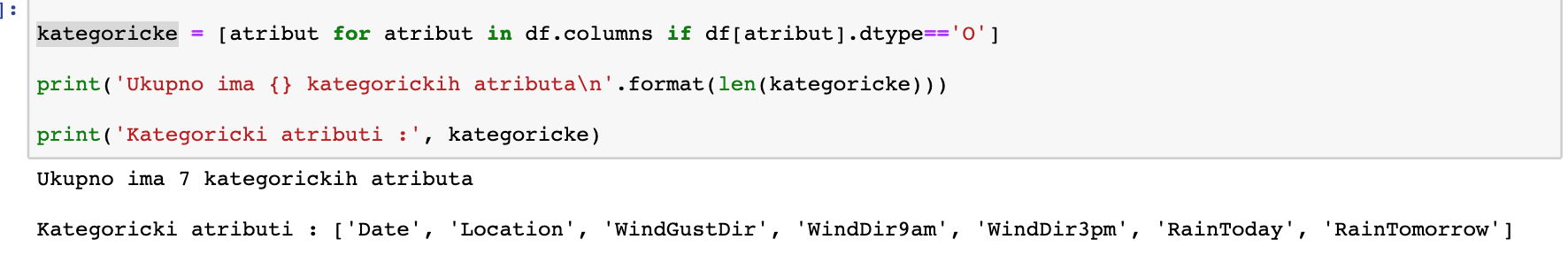


*Информације о скупу*

У скупу података постоји мешавина категоричких и нумеричких променљивих. Категоричке променљиве су типа *object*. Нумеричке променљиве имају формат података типа *float64*.

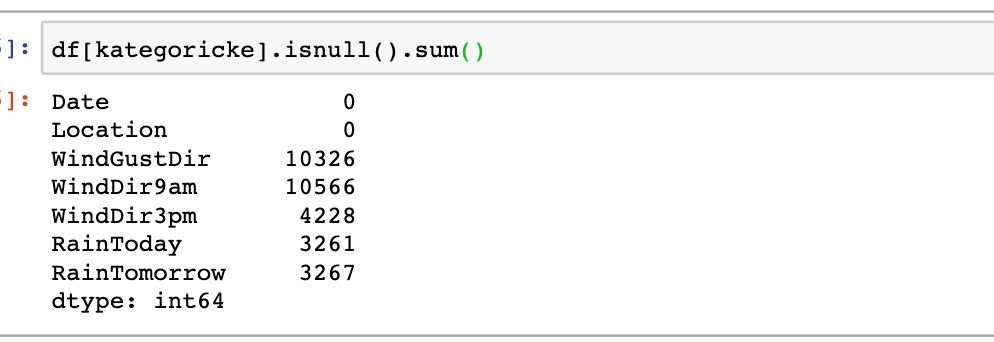
Уколико издвојимо само категоричке променљиве, укупан број атрибута је 7. Међутим, атрибут  “Date” је погрешно класификован. Уочава се и да постоје две

бинарне категоричке променљиве – *RainToday* и *RainTomorrow*, где је *RainTomorrow* циљни атрибут.

****

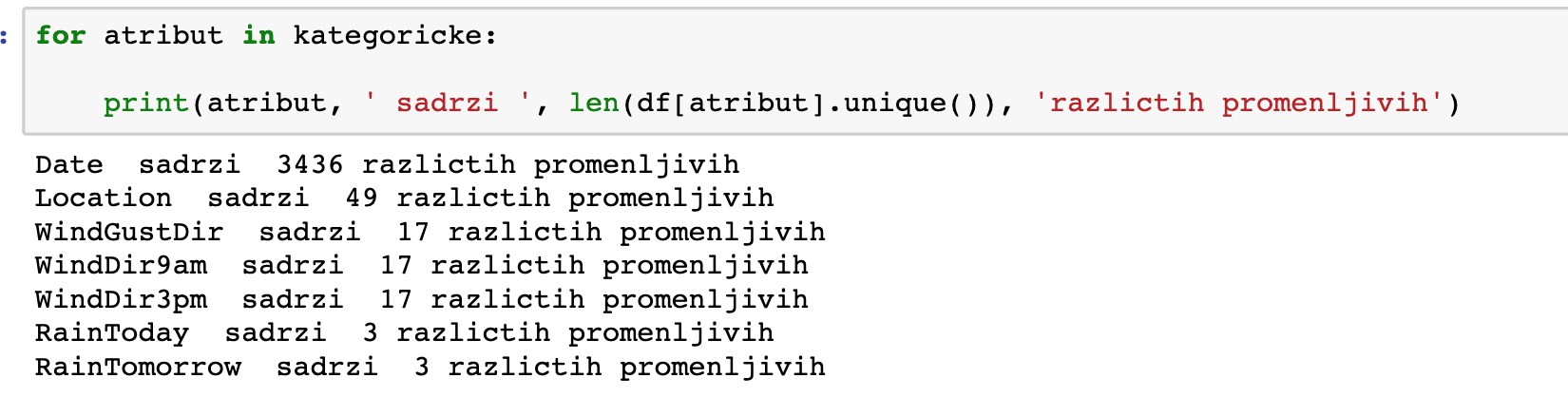
*Категорички атрибути*

Проверава се да ли категоричке променљиве садрже недостајуће вредности:



*Провера недостајућих вредности категоричких атрибута*

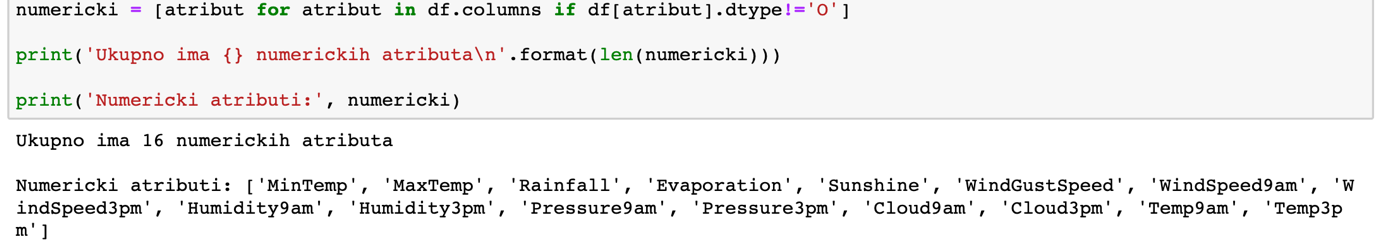
Потребно је проверити кардиналност променљивих и уколико је потребно извршити агрегацију. Агрегацијом се постиже стабилност података.



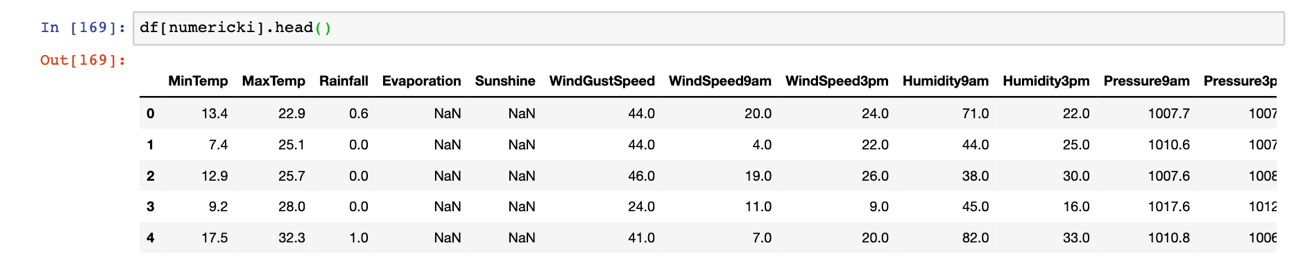
*Јединствене вредности категоричких атрибута*

Уочава се да се Date издваја од осталих и потребно је извршити агрегацију.

Издвајањем само нумеричких атрибута уочава се да има 16 нумеричких атрибута континуалног типа.

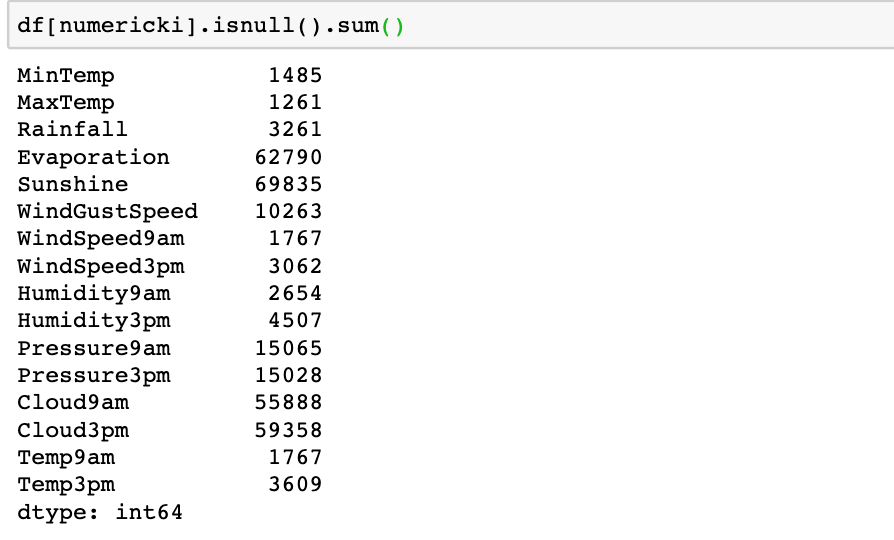


*Нумерички атрибути*



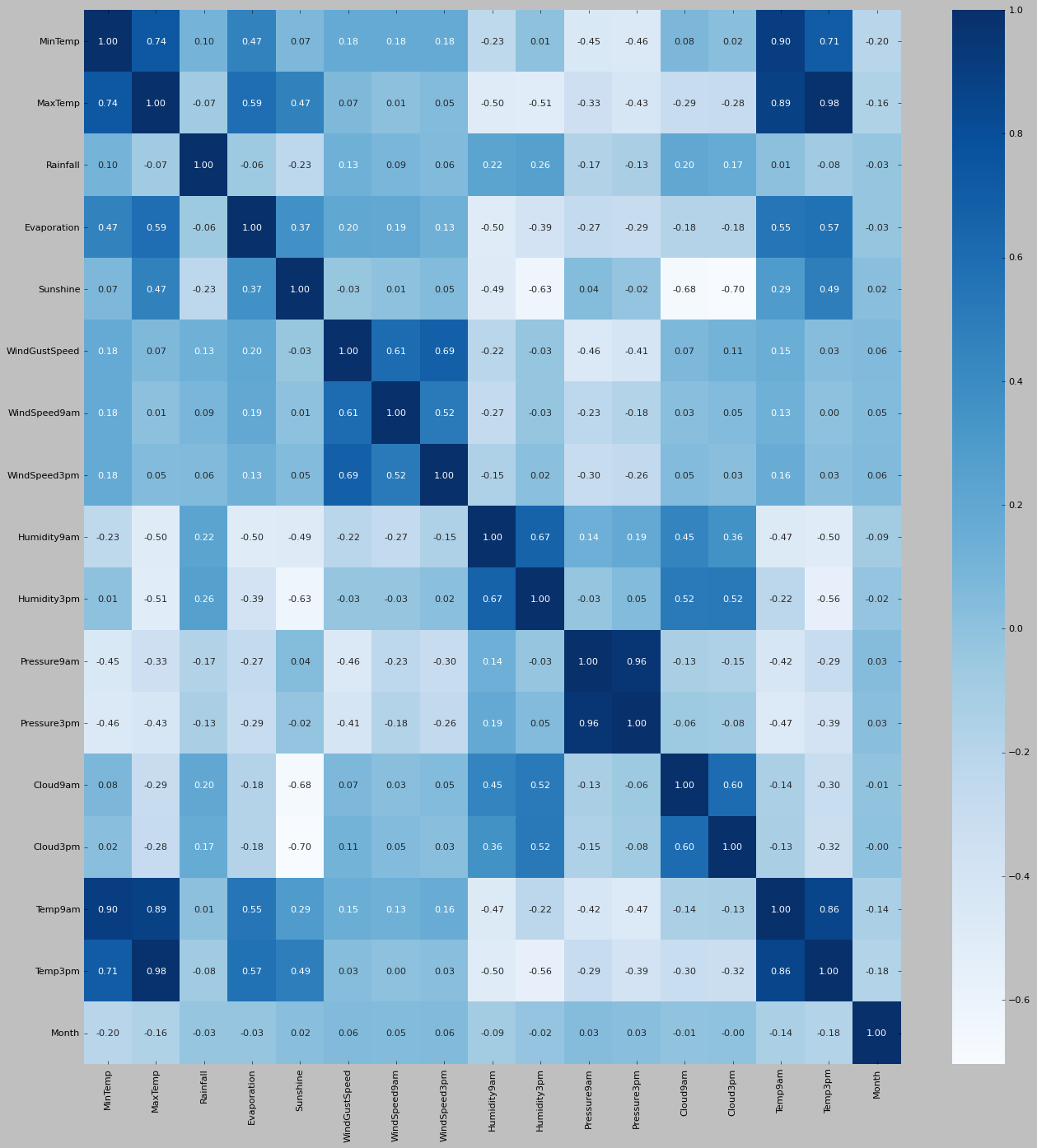
*Нумерички атрибути- статистика*

Проверава се да ли садрже недостајуће вредности:



*Провера недостајућих вредности нумеричких атрибута*

Како би најлакше увидели корелацију између атрибута, матрица корелација је приказана преко топлотних мапа.

****

*Матрица корелације*

Променљиве које су јако позитивно корелисане:

* WindGustSpeed и WindSpeed3pm (коефицијент корелације = 0,69),
* Pressure9am и Pressure3pm (коефицијент корелације = 0,96),
* Temp9am и Temp3pm (коефицијент корелације = 0,86),
* MinTemp и MaxTemp (коефицијент корелације = 0,74),
* MinTemp и Temp3pm (коефицијент корелације = 0,71),
* MinTemp и Temp9am (коефицијент корелације = 0,90),
* MaxTemp и Temp9am (коефицијент корелације = 0,89),
* MaxTemp и Temp3pm (коефицијент корелације = 0,98).

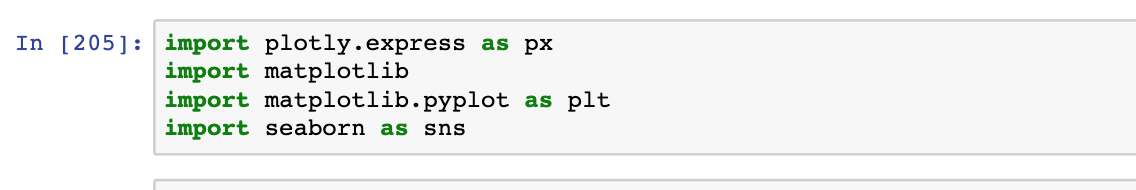
Позитивна корелација показује линеарни однос између неке две променљиве, када се једна променљива смањује како се друга променљива смањује, и обрнуто.

Негативне корелације имају следеће променљиве:

* Cloud3Pm Sunshine (коефицијент корелације = -0.70),
* Cloud9am Sunshine (коефицијент корелације = -0.68),
* Humidity Sunshine (коефицијент корелације = -0.63),
* Humidity3Pm Temp3Pm (коефицијент корелације = -0.56).

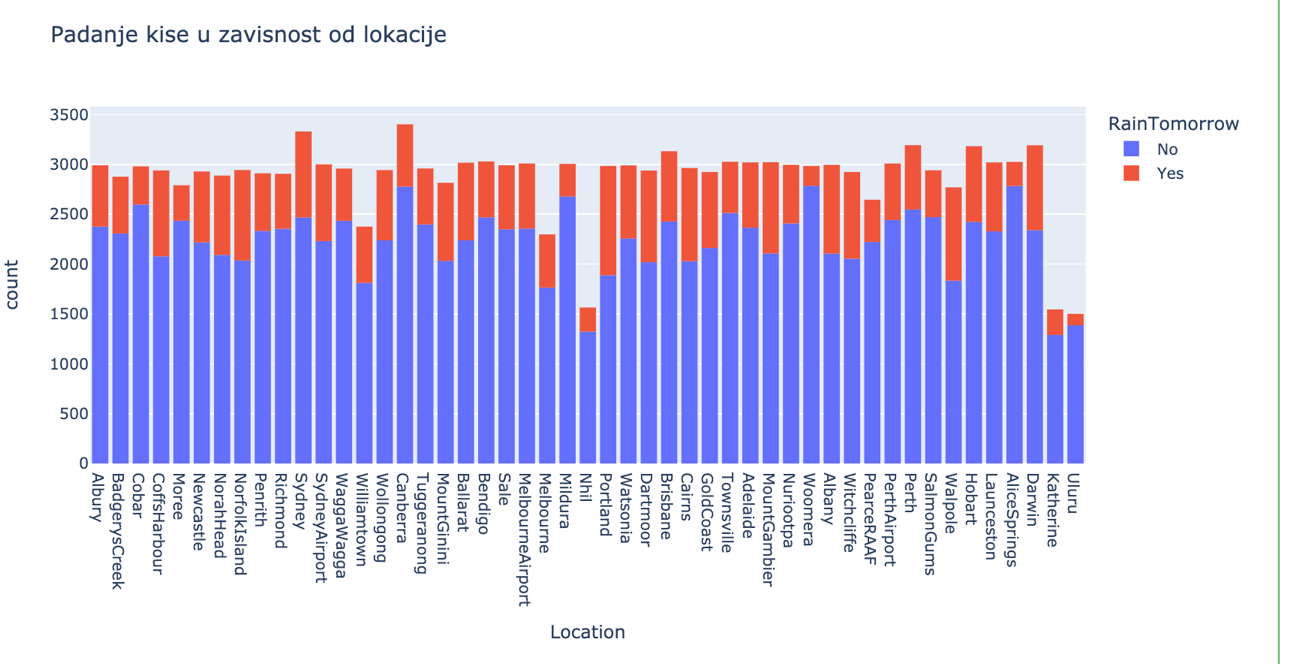
Негативна корелација је однос између две променљиве у којој се једна променљива повећава како се друга смањује, и обрнуто. Висока негативна корелација значи да је однос који постоји између две променљиве увек супротан.

Уз помоћ библиотеке *plotly.express* могуће је направити интерактивне графике.



*Библиотека plotly.express*

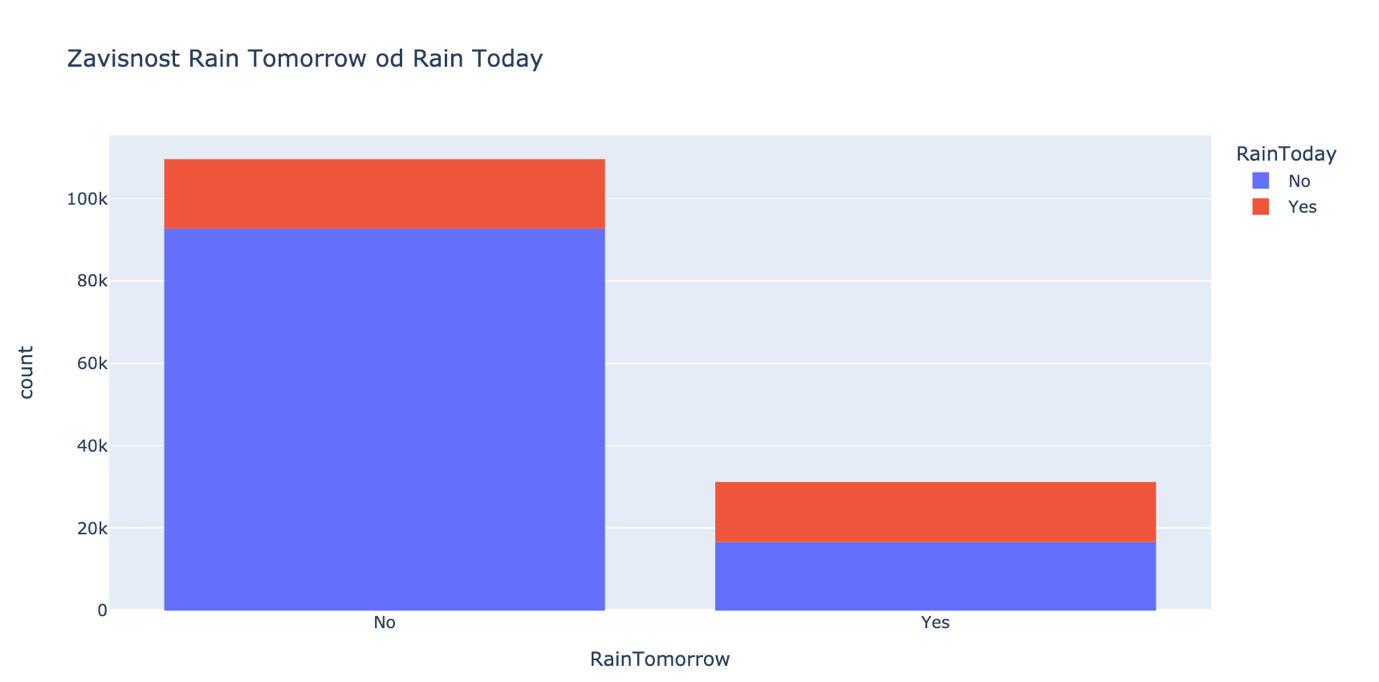
За потребе овог пројекта ова библиотека ће се користити за процењивање и визуелизацију односа међу атрибутима.



*Зависност Rain Tomorrow од Location*

Уколико посматрамо зависност циљног атрибута од локације уочава се да постоји униформна дистрибуцију података по локацији.

Нхил, и Улуру имају најмањи број падавина, док Portland бележи највише међу укупно 49 локација. Закључује се да је локација дефинитивно фактор који утиче да ли ће сутра падати киша.



*Зависност Rain Tomorrow од Rain Today*

Уочава се да немамо једнак број запажања у свакој класи. На основу хистограма закључује се да ако данас није падала киша, постоји велика шанса да ни сутра неће падати киша.

**Препроцесирање**

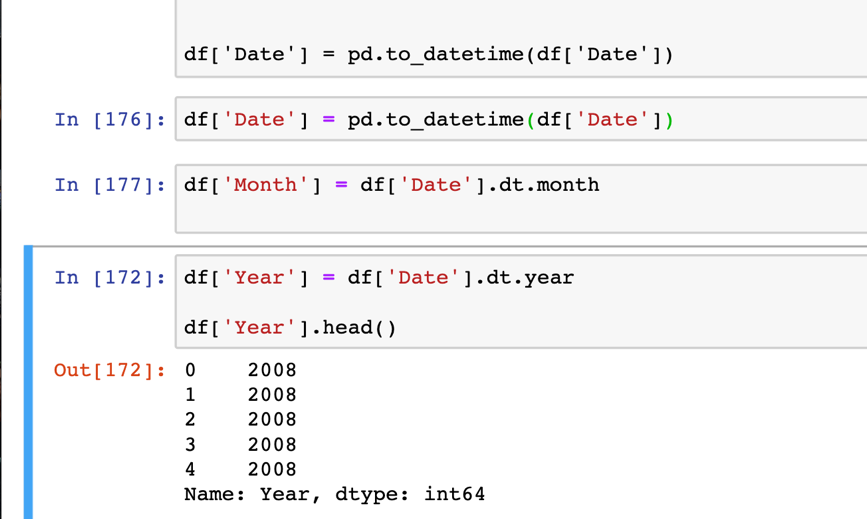
Сваки модел поступа на одређени начин са недостајућим подацима (*енгл.missing values*), и поједине технике моделирања су боље од других за рад са њима. Постоје различите технике које се користе за рад са недостајућим подацима, где неке подразумевају брисање недостајућих вредности, примене различитих алгоритама или процену вредности на основу средњих вредности, медијане..

Како овај скуп садржи и категоричке и нумеричке атрибуте, одлучено је да се недостајуће вредности не процењују, већ једноставно обришу. Функција која се користи за уклањање недостајућих вредности је *dropna()*. Ради провере да ли су остале неке недостајуће вредности користимо *df.isna().sum().sum()* где сумирамо све вредности које су нула по целом скупу.



*Провера да ли су остале недостајуће вредности*

Као што је у претходном поглављу поменуто, неопходно је извршити агрегацију променљивих у оквиру атрибута Date. Како се атрибут Date састоји од дана, месеца и године извршићемо промену скале тако да уместо свих дана у години издвају се месеци и године. Првобитно је неопходно претворити Date у одговарајући формат.



*Промена скале атрибута Date*

Након овога се уклања атрибут Date јер сада представља вишак у скупу.



*Брисање Date*

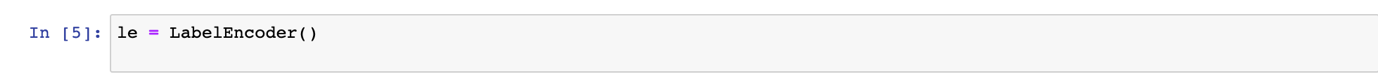
Прикупљени подаци су комбинација и категоричких и континуалних променљивих, већина алгоритама машинског учења неће разумети или неће моћи да се ради са категоричким променљивим. Потребно је конвертовати категоричке податке у нумеричке, односно извршити кодирање категоричких података.

За ове потребе коришћен је *LabelEncoding*, који категоричку вредност замењује нумеричком вредношћу у опсегу (0,n-1), где је n број класа. Како би се користило кодирање неопходно је укључити модел за препроцесирање из пакета *sklearn.*



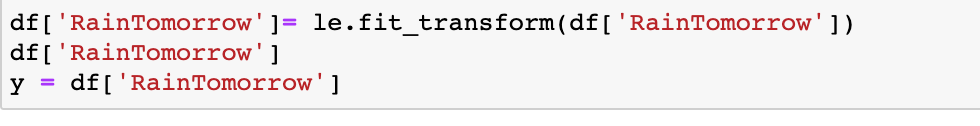
*Библиотека за препроцесирање*

Прво је потребно креирати инстанцу *LabelEncoder()* која се чува у променљивој *le*.

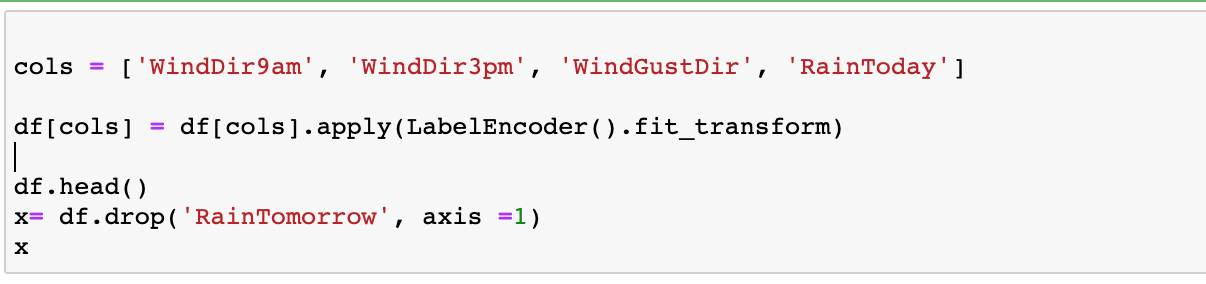


*LabelEncoder()*

Након тога, коришћењем методе *le.fit\_transform()* претварамо категоричку променљиву у нумеричку. На слици је приказано како се то ради са циљним атрибутом *y*.

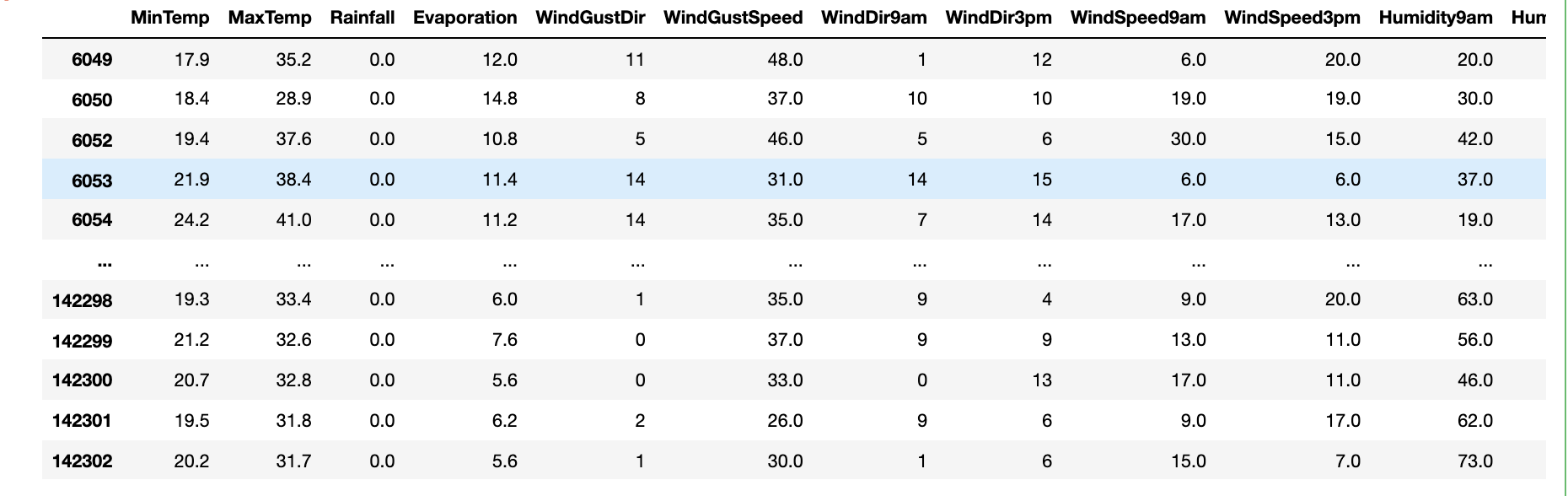
*Кодирање циљног атрибута*

Потребно је кодирати и остале категоричке вредности.



*Кодирање осталих атрибута*

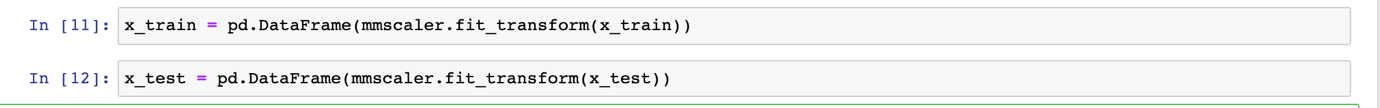
Након кодирања x изгледа овако:



Како се на основу слике(…) може уочити да су вредности различитих опсега, те како један атрибут не би био утицајнији од другог потребно је скалирати дате вредности. Најпре, је потребно креирати инстанцу *MinMaxScaler()* која се чува у променљивој *mmscaler*.



Скалирају се вредности улаза на тренинг и тест скупу:



Након одговарајућих техника препроцесирања подаци су сада спремни за одговарајуће алгоритме класификације.

**Алгоритми класификације:**

Не постоjи алгоритам за коjи може да се каже да jе ’наjбољи’ у свим случаjевима. Избор алгоритма зависи од конкретног проблема, врсте података, од жељеног начина приказа резултата, итд. За конкретан проблем најбоље је испробати различите алгоритме и упоредити њихове резултате.

Следећа функција ће бити од помоћи приликом тумачења и упоређивања различитих алгоритама. Приликом позива функције class\_info() доступне су информације о матрицама конфузије, извештајима класификације, прецизностима на тренинг и тест делу.

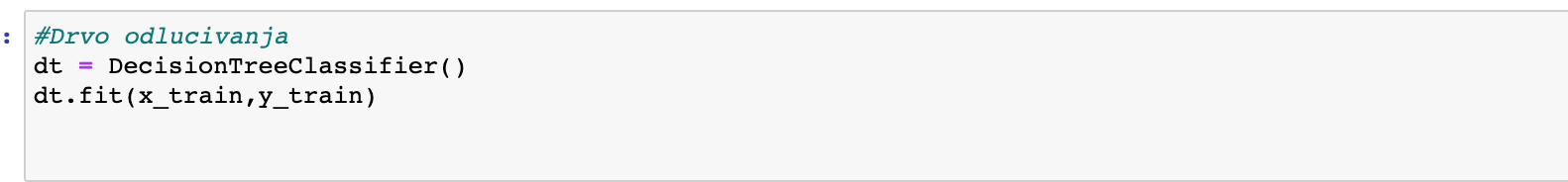


**Дрво одлучивања**

Како би се користио овај алгоритам у програмском језику Python, неопходно је укључити модул који укључује моделе засноване на дрвету одлучивања за класификацију и регресију.

****

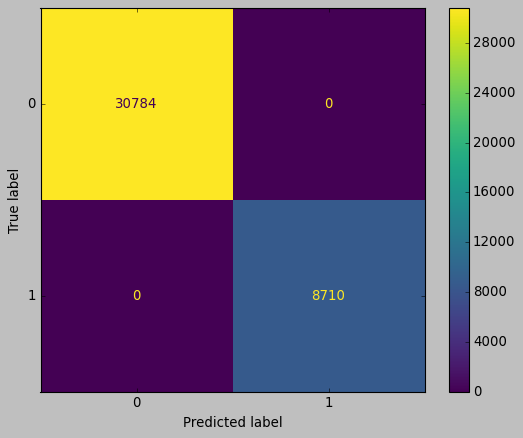
Након тога потребно је креирати класификатор и обучити модел на основу обучавајућих парова.

****

Након обучавања позвана је функција *class\_info()*, како би се добиле све битне метрике за један алгоритам.

**Тренинг скуп:**

За тренинг скуп добијена је матрица конфузије:



Може се закључити да овај алгоритам савршено класификује вредности у тренинг скупу тј. све вредности су исправно класификоване.

Алгоритам можемо и оценити уз помоћ извештаја класификације који пружа информације о одзиву, прецизности, Ф1 мери и подршци. У следећој табели приказан је извештај класификације за тренинг скуп.

Извештај класификације тренинг скупа:

precision recall f1-score support

0 1.00 1.00 1.00 30784

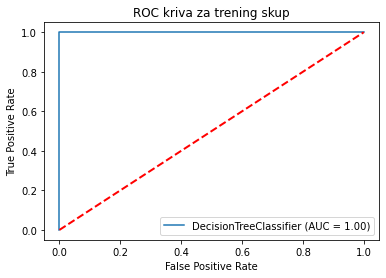
1 1.00 1.00 1.00 8710

accuracy 1.00 39494

macro avg 1.00 1.00 1.00 39494

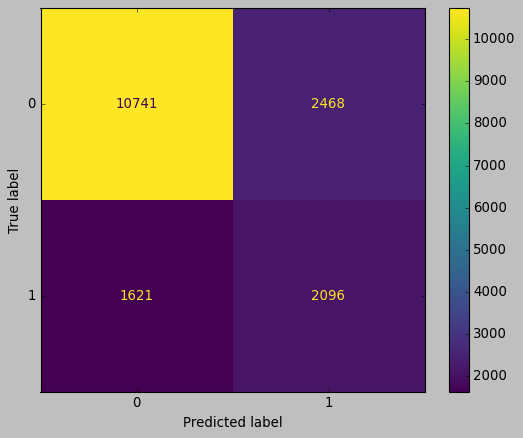
weighted avg 1.00 1.00 1.00 39494

ROC крива за тренинг скуп показује да се ради о идеалном случају, где је површина испод криве (енгл. AUC-Area under curve) једнака 1. Што је ROC крива ближа горњем левом углу, модел је ефикаснији.



**Тест скуп:**

За тест скуп матрица конфузије изгледа:



На основу матрице конфузије види се да у тест скупу алгоритам погрешно класификује циљне класе и то 1621 пута је погрешно предвиђено да ће падати киша, а 2468 пута је погрешно предвиђено да неће падати киша.

Извештај класификације тест скупа:

precision recall f1-score support

0 0.87 0.81 0.84 13209

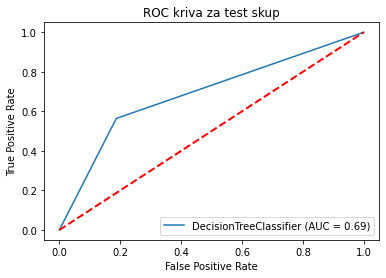
1 0.46 0.56 0.51 3717

accuracy 0.76 16926

macro avg 0.66 0.69 0.67 16926

weighted avg 0.78 0.76 0.77 16926

На основу ROC криве и површине испод ње види се да је класификација у тест скупу гора него у тренинг скупу. Вредност површине испод криве је 0.69, док у тренинг скупу она износи 1. Такође сама крива је ближа криви безвредног теста (дијагонална црвена линија). Крива безвредног теста укључује тачку са 50 % осетљивости и 50 % специфичности.



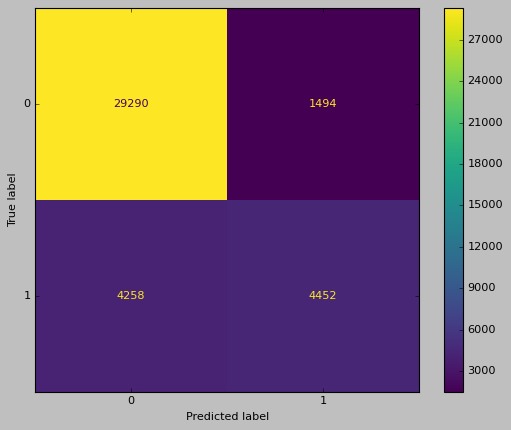
На основу анализа уз помоћ различитих метрика закључује се да је дошло до преприлагођавања при коришћењу Дрвета одлучивања.

Конструисањем сложених модела, овај алгоритам има тенденцију ка преприлагођавању податка. Овакви модели највероватније неће вршити добру генерализацију са невиђеним подацима. Једно од могућих решења за овај проблем је пре-поткресивање и пост-поткресивање. Пре-поткресивање значи ограничавање дубине стабла пре стварања (задавањем параметара као што су највећа дубина стабла, минимални број узорака потребан да би се чвор наставио делити и минимални број инстанци потребних за лист), док се пост-поткресивањем уклањају неинформативни чворови након што је дрво изграђено. Приликом коришћења *sklearn* библиотеке врши се само пре-поткресивање.

Такође, како би смањили шансе за преприлогођавање може се користити унакрсна валидација.

**Тренинг скуп:**

Уколико се користи унакрсна валидација добијају се другачије вредности за матрицу конфузије. На основу матрице конфузије види се да у тренинг скупу алгоритам погрешно класификује циљне класе и то 4258 пута је погрешно предвиђено да неће падати киша, а 1494 пута је погрешно предвиђено да ће падати киша.



Како долази до погрешног класификовања, очекивано је да ће се сада бити мања прецизност, одзив и Ф1 мера.

Извештај класификације тренинг скупа:

precision recall f1-score support

0 0.87 0.95 0.91 30784

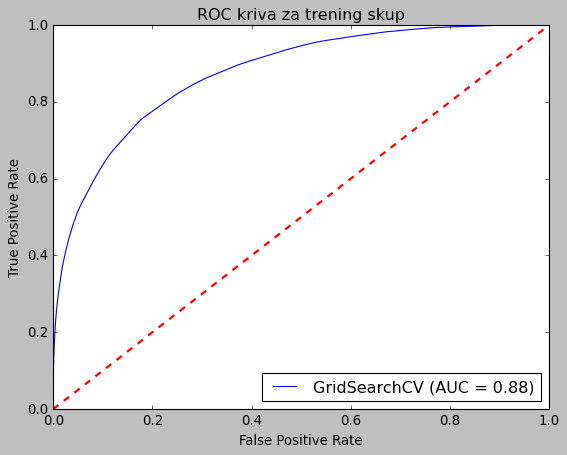
1 0.75 0.51 0.61 8710

accuracy 0.85 39494

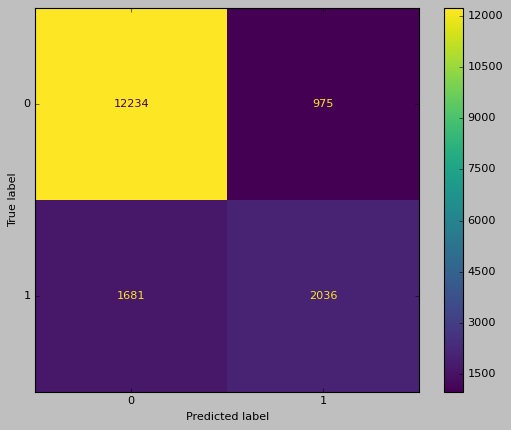
macro avg 0.81 0.73 0.76 39494

weighted avg 0.85 0.85 0.84 39494

ROC крива више није идеална као у претходном поступку, међутим оваква крива показује да је модел добар.



**Тест скуп:**



У поређењу са тренинг скупом добија се већа вредности прецизност за 0-ту класу, док се веће вредности прве класе добијају за одзив и Ф1 меру.

Извештај класификације тест скупа:

precision recall f1-score support

0 0.88 0.93 0.90 13209

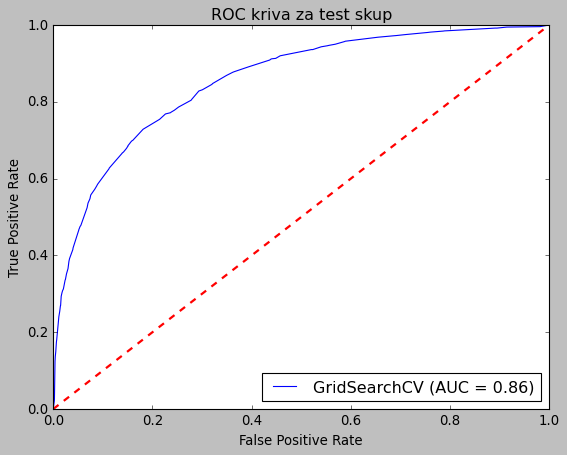
1 0.68 0.55 0.60 3717

accuracy 0.84 16926

macro avg 0.78 0.74 0.75 16926

weighted avg 0.83 0.84 0.84 16926

На основу ROC криве и површине испод ње види се да је класификација у тест скупу јако мало разликује у односу на тренинг скупу. Вредност површине испод криве је 0.86, док у тренинг скупу она износи 0.88.



Унакрсном валидацијом су добијени бољи резултати и боља прецизност самог алгоритма.

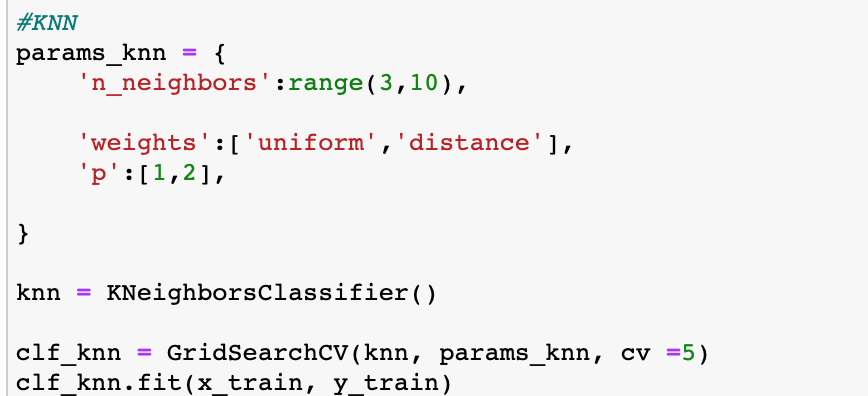
**К најближих суседа**

Како би се користио овај алгоритам у програмском језику Python, неопходно је укључити модул који укључује моделе засноване на к најближих суседа за класификацију.

****

За овај алгоритам коришћена је унакрсна валидација, те је потребно дефинисати параметре алгоритма унапред. Одређено је да број суседа варира од 3-10, тежина суседа може бити униформна и обрнута растојању, за растојање је задато да буде Еуклидско и Менхетн.

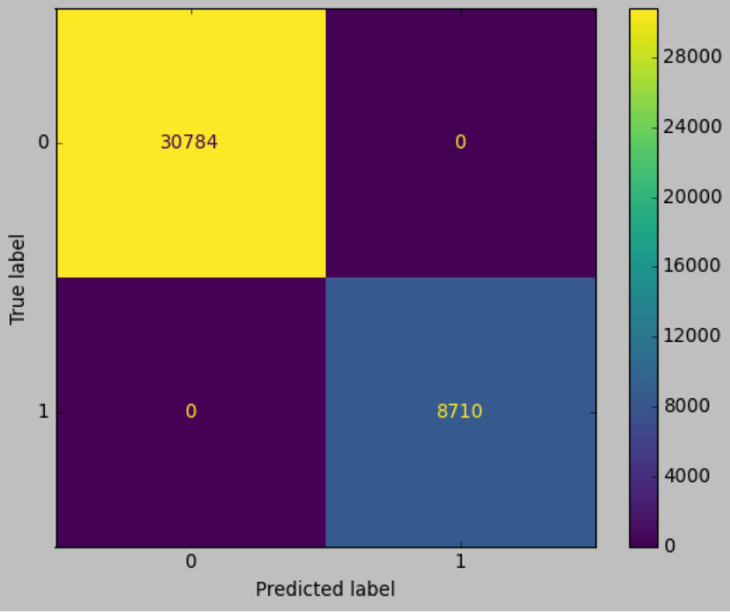
Након тога потребно је креирати класификатор и обучити модел на основу обучавајућих парова.

****

Након обучавања позвана је функција *class\_info()*, како би се добиле све битне метрике за један алгоритам.

**Тренинг скуп:**

За тренинг скуп добијена је матрица конфузије:

****

Може се закључити да овај алгоритам савршено класификује вредности у тренинг скупу тј. све вредности су исправно класификоване.

Алгоритам можемо и оценити уз помоћ извештаја класификације који пружа информације о одзиву, прецизности, Ф1 мери и подршци. У следећој табели приказан је извештај класификације за тренинг скуп.

Извештај класификације тренинг скупа:

precision recall f1-score support

0 1.00 1.00 1.00 30784

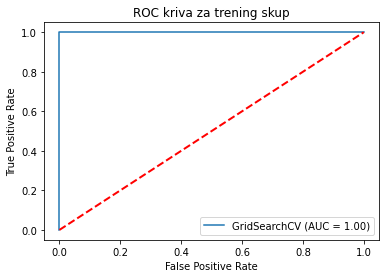
1 1.00 1.00 1.00 8710

accuracy 1.00 39494

macro avg 1.00 1.00 1.00 39494

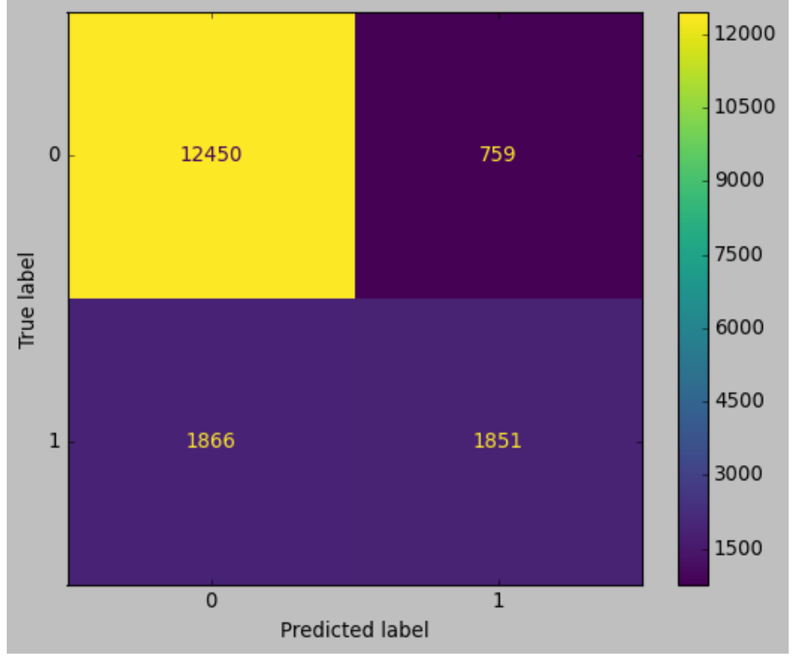
weighted avg 1.00 1.00 1.00 39494

ROC крива за тренинг скуп показује да се ради о идеалном случају, где је површина испод криве (енгл. AUC-Area under curve) једнака 1.



**Тест скуп:**

Матрица конфузије тест скупа је:



На основу матрице конфузије види се да у тест скупу алгоритам погрешно класификује циљне класе и то 1866 пута је погрешно предвиђено да неће падати киша, а 759 пута је погрешно предвиђено да ће падати киша.

На основу извештаја класификације уочава се да:

* Прецизност 0-те класе је 0.87, док је прецизност прве 0.71
* Одзив нулте класе је већи и износи 0.94, док је одзив прве 0.50
* Ф1 мера је мања код прве класе и износи 0.59
* Целокупна прецизност модела је 0.84

Извештај класификације тест скупа:

precision recall f1-score support

0 0.87 0.94 0.90 13209

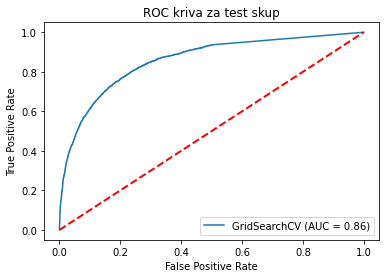
1 0.71 0.50 0.59 3717

accuracy 0.84 16926

macro avg 0.79 0.72 0.74 16926

weighted avg 0.83 0.84 0.83 16926

На основу ROC криве и површине испод ње види се да је класификација у тест скупу гора него у тренинг скупу. Вредност површине испод криве је 0.86, док у тренинг скупу она износи 1.



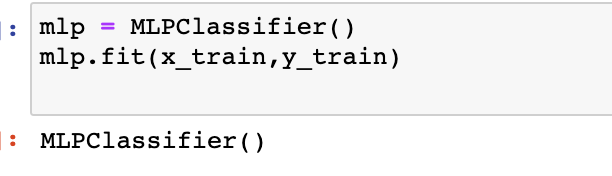
На основу добијених резултата може се закључити да је дошло до преобучавања приликом коришћења овог алгоритма.

**MLP- Вишеслојни перцептрон**

Да би се омогућило да се користе неуронске мреже у програмском језику Python потребно је активирати модул који укључује моделе засноване на неуронским мрежама. У овом случају користи се вишеслојни перцептрон.

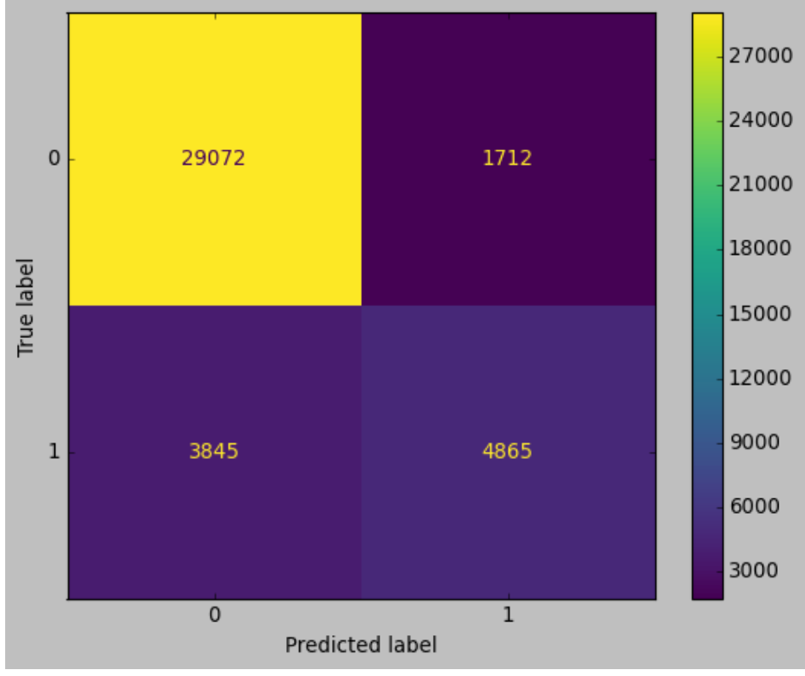
****

Потом следи креирање класификатора и обучавање модела на основу обучавајућих парова.

****

**Тренинг скуп:**

На основу добијене матрице конфузије уочава се да приликом коришћења овог алгоритма долази до грешке у класификацији и при тренирању. Наиме, 3845 пута је погрешно класификовано да неће падати киша, а 1712 пута је погрешно класификовано да ће падати.

****

Очекивано се добијају мање вредности за прецизност, одзив и Ф1 меру у односу на претходне алгоритме.

Извештај класификације тренинг скупа:

precision recall f1-score support

0 0.88 0.94 0.91 30784

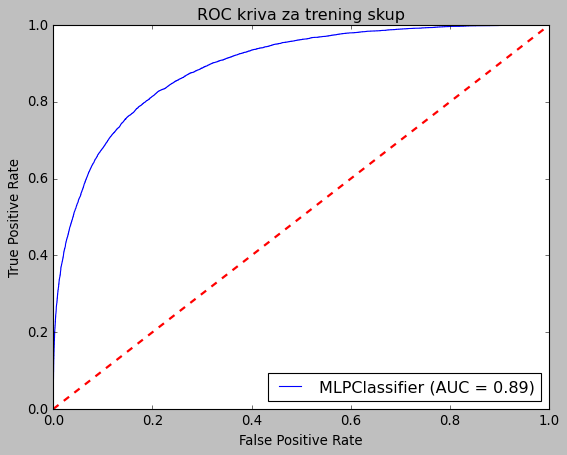
1 0.74 0.56 0.64 8710

accuracy 0.86 39494

macro avg 0.81 0.75 0.77 39494

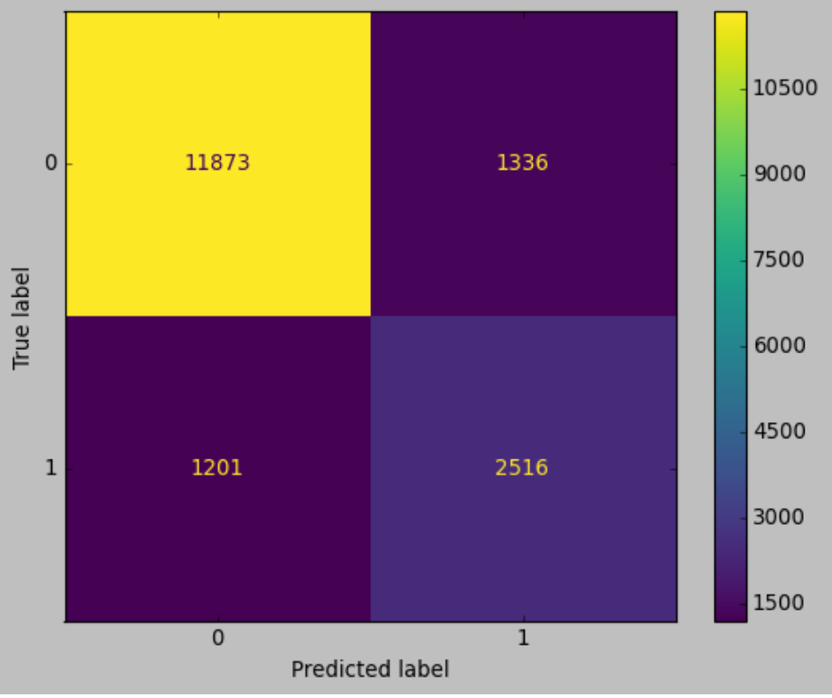
weighted avg 0.85 0.86 0.85 39494

Примећује се да ROC крива није идеална, међутим и даље се може сматрати да овај тест даје ваљане резултате. Површина испод криве је 0.89.



**Тест скуп:**

На основу добијене матрице конфузије уочава се да приликом коришћења овог алгоритма долази до грешке у класификацији. Наиме, 1201 пута је погрешно класификовано да неће падати киша, а 1336 пута је погрешно класификовано да ће падати.



У поређењу са тренинг скупом добија се већа вредности прецизност за 0-ту класу, док се веће вредности прве класе добијају за одзив и Ф1 меру.

Извештај класификације тест скупа:

precision recall f1-score support

0 0.91 0.90 0.90 13209

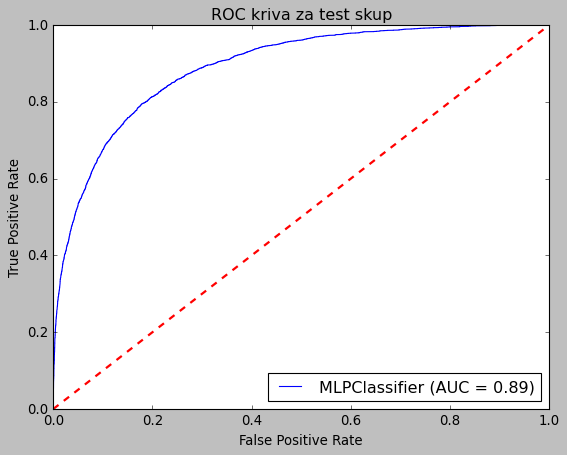
1 0.65 0.68 0.66 3717

accuracy 0.85 16926

macro avg 0.78 0.79 0.78 16926

weighted avg 0.85 0.85 0.85 16926

Површина испод криве за тест скуп је иста као и за тренинг скуп. Закључује се да се ROC  кривом не може увидети који модел је бољи јер дају идентичне криве.



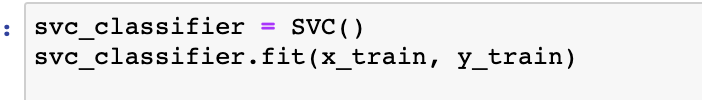
Прецизност тест и тренинг скупа је веома блиска на основу тога и на основу осталих метрика може се закључити да није дошло до преприлагођавања.

**Support Vector Machine**

За потребе коришћење потпорних вектора у Python-у потребно је активирати модул који укључује моделе засноване на **SVM**.

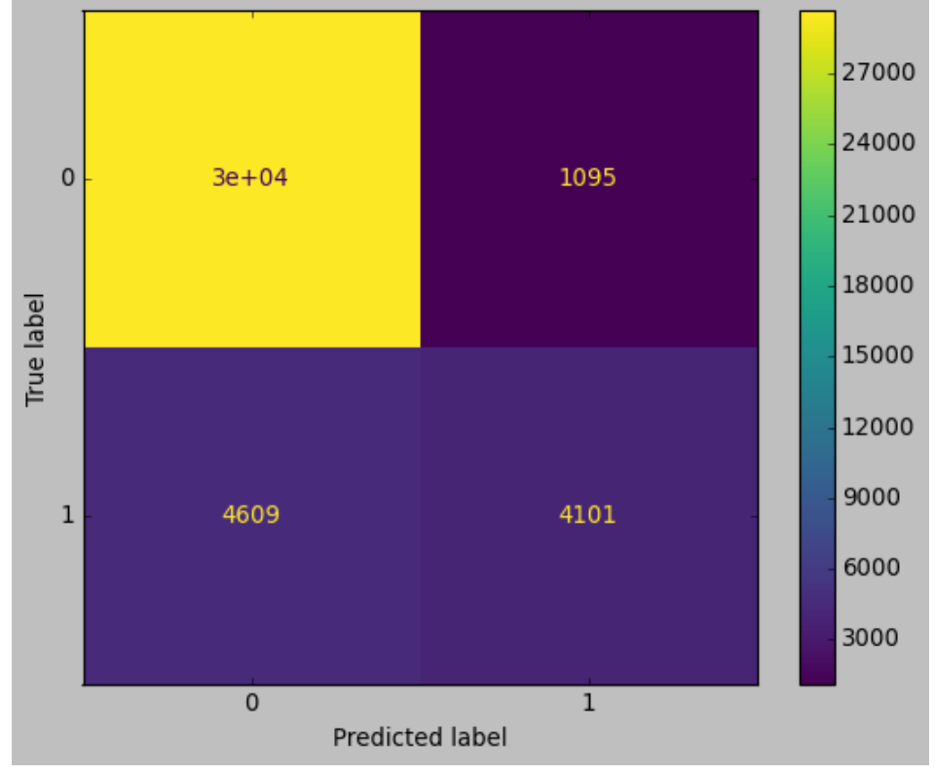
****

Потом следи креирање класификатора и обучавање модела на основу обучавајућих парова.



**Тренинг скуп:**

На основу добијене матрице конфузије уочава се да приликом коришћења овог алгоритма долази до грешке у класификацији. Наиме, 4609 пута је погрешно класификовано да неће падати киша, а 1095 пута је погрешно класификовано да ће падати.



Извештај класификације тренинг скупа:

precision recall f1-score support

0 0.87 0.96 0.91 30784

1 0.79 0.47 0.59 8710

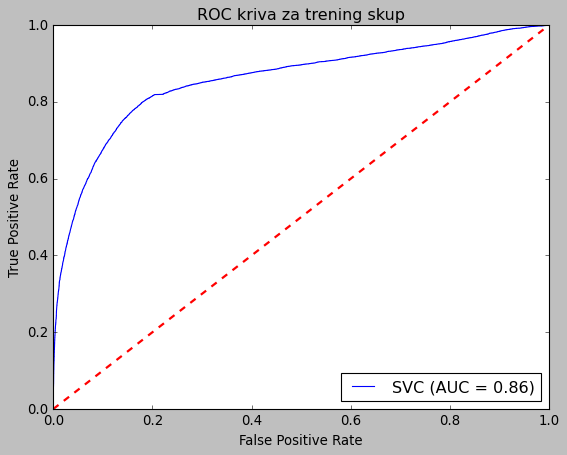
accuracy 0.86 39494

macro avg 0.83 0.72 0.75 39494

weighted avg 0.85 0.86 0.84 39494

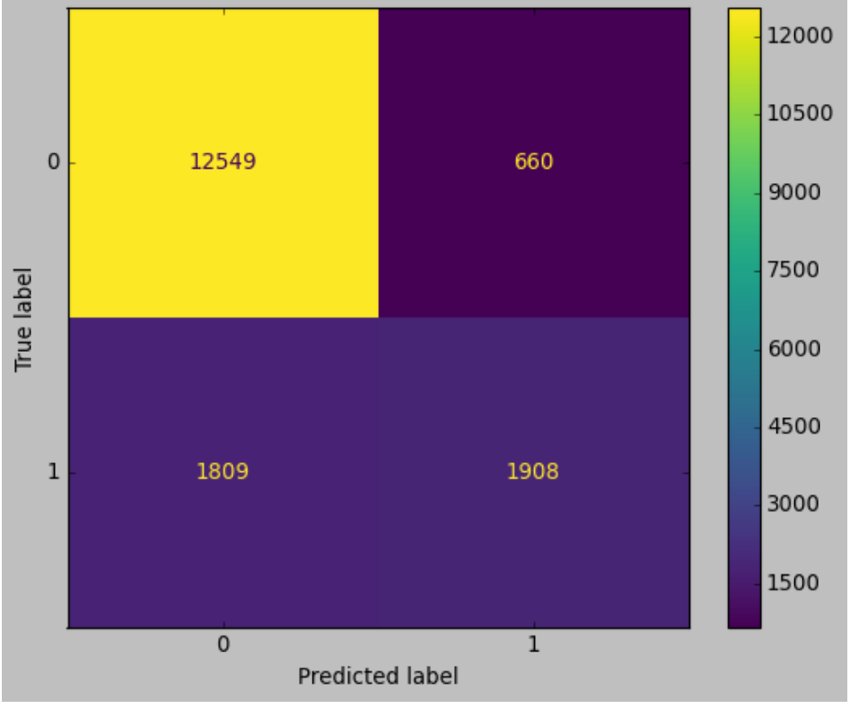
На основу извештаја класификације уочава се да:

* Прецизност 0-те класе је 0.87, док је прецизност прве 0.79
* Одзив нулте класе је већи и износи 0.96
* Ф1 мера је мања код прве класе и износи 0.59
* Целокупна прецизност модела је 0.86



**Тест скуп:**

На основу добијене матрице конфузије уочава се да приликом коришћења овог алгоритма долази до грешке у класификацији. Наиме, 1809 пута је погрешно класификовано да неће падати киша, а 660 пута је погрешно класификовано да ће падати.



На основу извештаја класификације уочава се да:

* Прецизност 0-те класе је 0.87, док је прецизност прве 0.74,
* Одзив нулте класе је већи и износи 0.95,
* Ф1 мера је мања код прве класе и износи 0.61,
* Целокупна прецизност модела је 0.85.

Добијено је да одзив и Ф1 мера имају већу вредност у тест скупу него у тренинг скупу за 1. класу.

Извештај класификације тест скупа:

precision recall f1-score support

0 0.87 0.95 0.91 13209

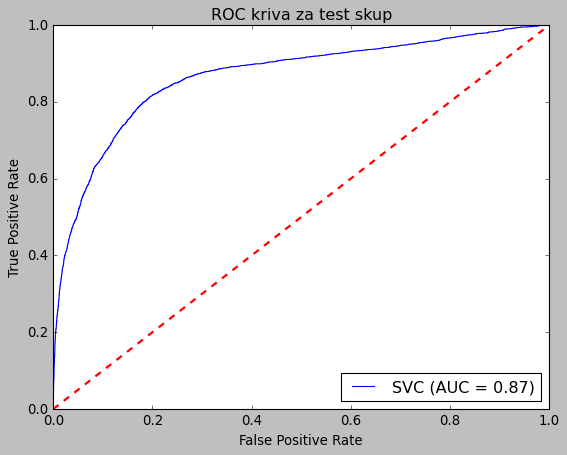
1 0.74 0.51 0.61 3717

accuracy 0.85 16926

macro avg 0.81 0.73 0.76 16926

weighted avg 0.85 0.85 0.84 16926

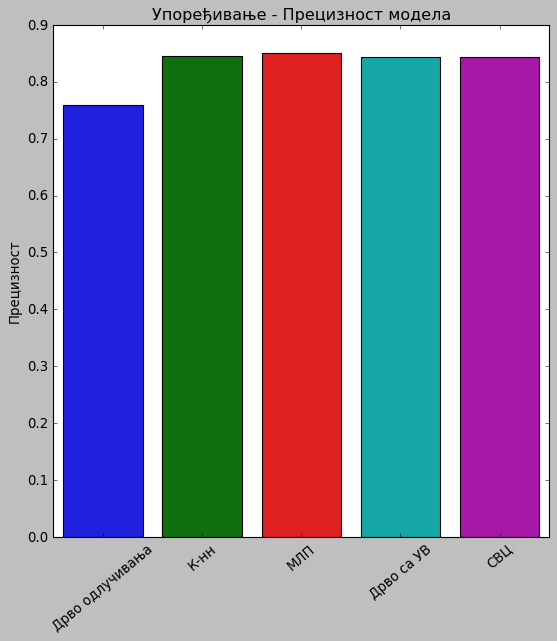
Може уочити да је површина испод криве за тест скуп већа него у тренинг скупу. Што би значило да се модел боље понаша на тест скупу.

****

Разлика између прецизност тест и тренинг скупа је веома мала, добијене су сличне вредности за одзив, Ф1 меру, као и за ROC криву и површину испод ње. Модел даје сличне резултате приликом коришћење тренинг и тест скупа, те се може закључити да је модел добар и да није дошло до преприлагођавања.

**Упоређивање модела у Python-у:**

Као мера за упоређивање различитих алгоритама класификације коришћена је прецизност. На основу приказане слике може се уочити да сви модели имају јако сличну вредност прецизности осим модела који користи дрво одлучивања (његова прецизност је 0.76).



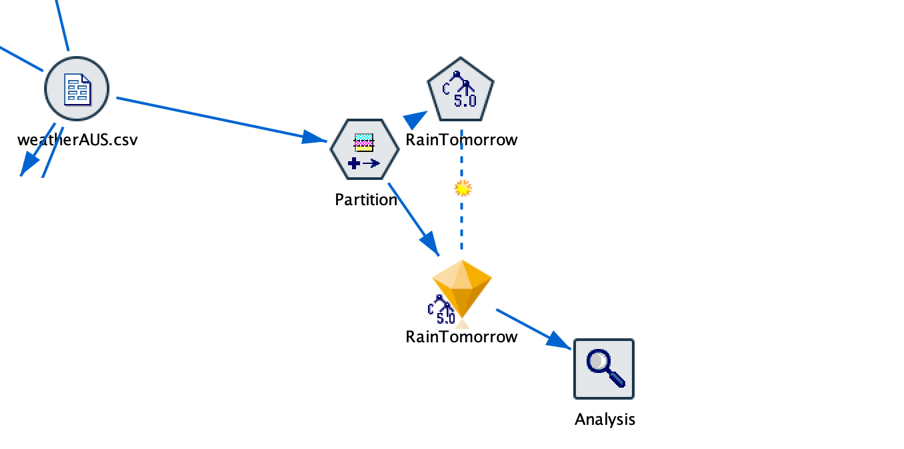
*Упоређивање алгоритама*

Алгоритми се могу поредити и на основу површине испод ROC криве. С тим критеријумом најбољи алгоритам је вишеслојни перцептрон.

Код SVM и MLP обучавање је вршено са default-ним параметрима јер је у супротном трајало превише дуго. Један од начина како би то могло да се реши је да се скрати скуп података.

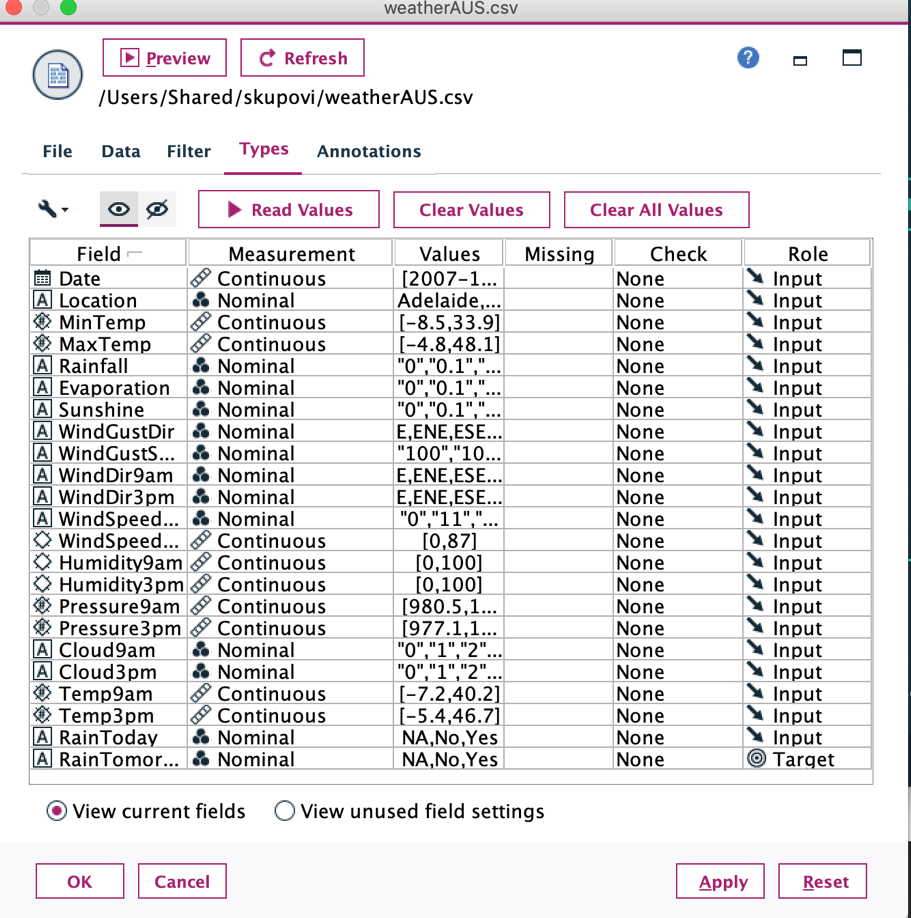
**Алгоритам C5.0 у SPSS-у:**

Пошто су сви остали алгоритми обрађени у програмском језику Python, изабрано је да ће се користити алгоритам C5.0.



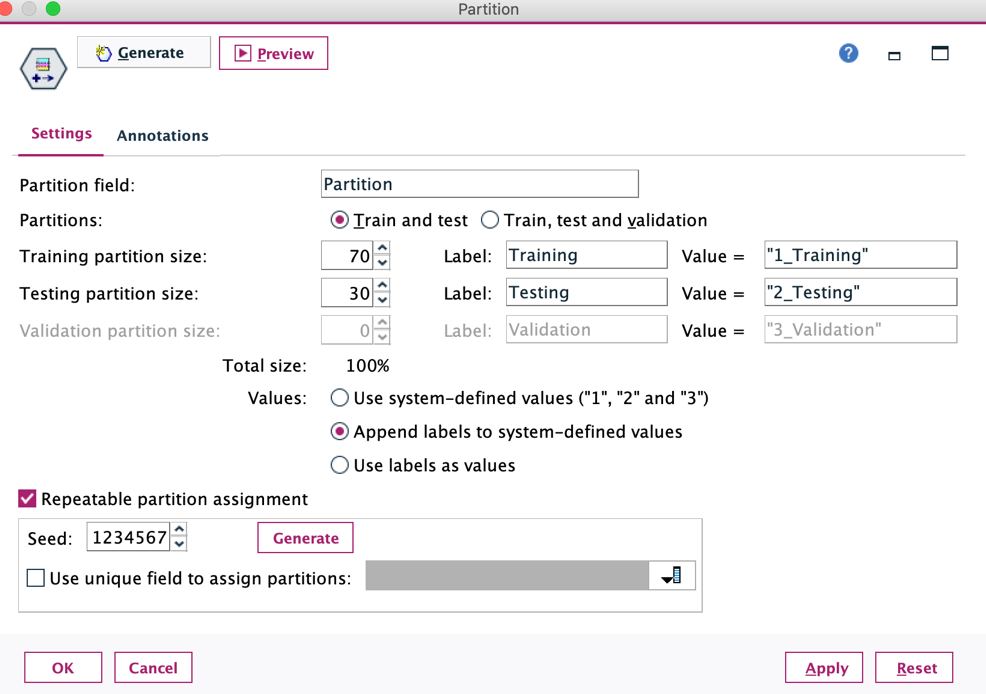
*SPSS modeler*

Најпре је потребно учитати скуп података и подесити циљни атрибут.



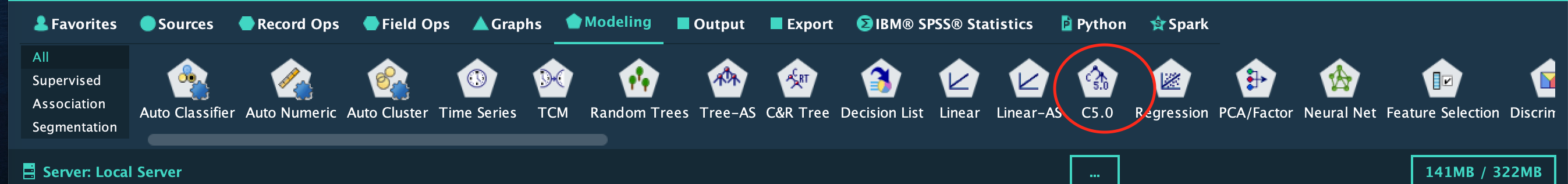
*Учитавање скупа*

Како је реч о класификацији потребно је поделити скуп на тренинг и тест део.

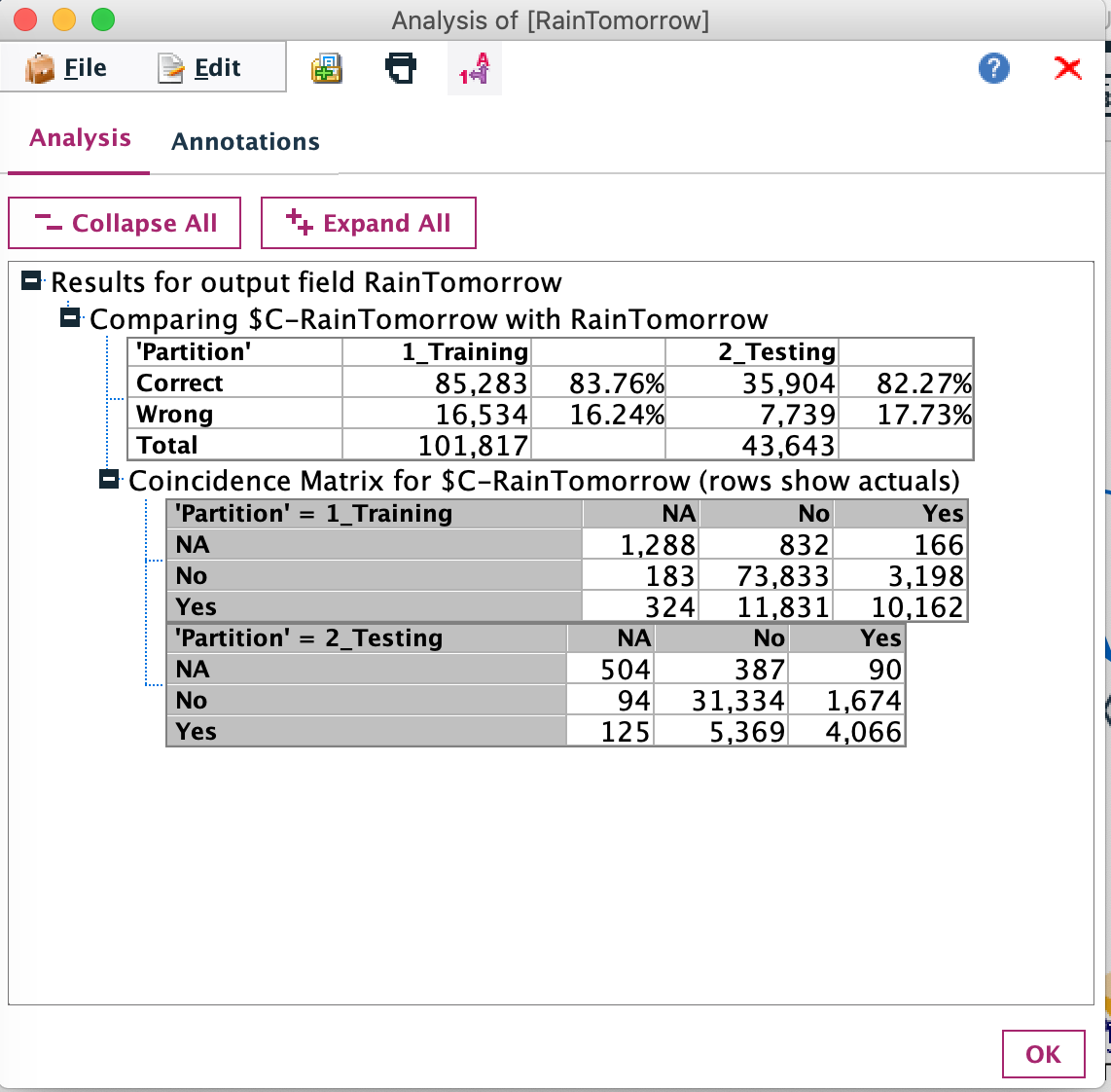


*Подела скупа*

Потом се бира одговарајући алгоритам, након покретања алгоритма добија се модел. Користећи чвор Analysis може се видети матрица конфузије тренинг и тест скупа.



*Избор алгоритма*



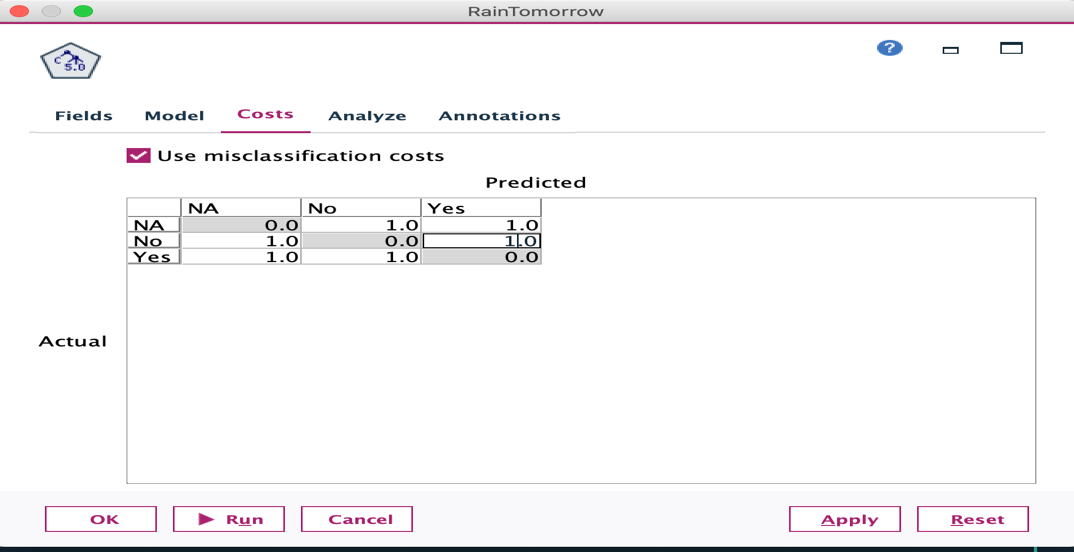
*Матрица конфузије*

Уочава се да добро класификовано 83.76% тренинг скупа односно 85,283 инстанци је добро класификовано од укупно 101,817. У тест скупу 82.27% инстанци је добро класификовано од укупно 43.643.

Посматрајући матрицу конфузије уочава се да долази до грешке класификације обе класе и у тренинг и у тест скупу. У тренинг скупу за 11,831 инстанци је погрешно препознао да неће падати киша, веће грешке прави за класу Yes.

У тест скупу је погрешно препознао за 5,369 инстанци да неће падати кипа, уочава се и да у тренинг скупу веће грешке прави за класу Yes.

Могуће решење за овај проблем би био да се постави већа цена када класа Yes погрешно класификује као No. То се ради у чвору самог алгоритма.



Међутим, када је цена класе Yes у NO повећана на 5.0, долази до преобучавања и незадовољавајућих резултата. Потребно је даље подешавање параметара и додатно препроцесирање података.

