2.Domaći zadatak – Stabla odlučivanja

Cilj klasifikacije: Za zadati skup slogova oblika (X, Y), gde je: X - skup atributa, Y - specijalni atribut koji određuje klasu, naći klasifikacioni model (funkciju) koji preslikava svaki skup atributa X u jednu od predefinisanih oznaka klasa Y.

Jedna od tehnika klasifikacije jeste Stablo odlučivanja (Stablo odluke- Decision tree -DT). Ideja kod kreiranja stabla je sledeća: Definisati pitanje kojim se heterogeni skup uzoraka deli na manje heterogene podskupove i to pitanje pridružiti jednom čvoru u stablu odlučivanja .Postupak ponoviti na svaki podskup uzoraka dobijen odgovorom na prethodno postavljeno pitanje i tako definisati čvorove odluke na sledećem nivou. Postupak treba ponavljati dok uzorak ne postane homogen (svi uzorci pripadaju istoj klasi) ili dok atributi svih objekata u uzorku ne postanu jednaki

Za potrebe domaćeg zadatka korišćen je sledeći skup podataka (u .csv formatu), preuzet sa sajta [https://www.kaggle.com/nareshbhat/wine-quality-binary-classification?select=wine.csv.](https://www.kaggle.com/nareshbhat/wine-quality-binary-classification?select=wine.csv)

Skup podataka sadrži atribute vezane za različite vrste vina. Treba odrediti kvalitet vina- dali je vino dobro ili loše.

Skup podataka sadrži 1599 zapažanja. Takođe sadrži 11 atributa i jedan atribut koji će predstavljati klasu.

Atributi:

1 - fiksna kiselost

2 - isparljiva kiselost

3 - limunska kiselina

4 - zaostali šećer

5 - hloridi

6 - slobodni sumpor-dioksid

7 - ukupni sumpor-dioksid

8 - gustina

9 - pH vrednost

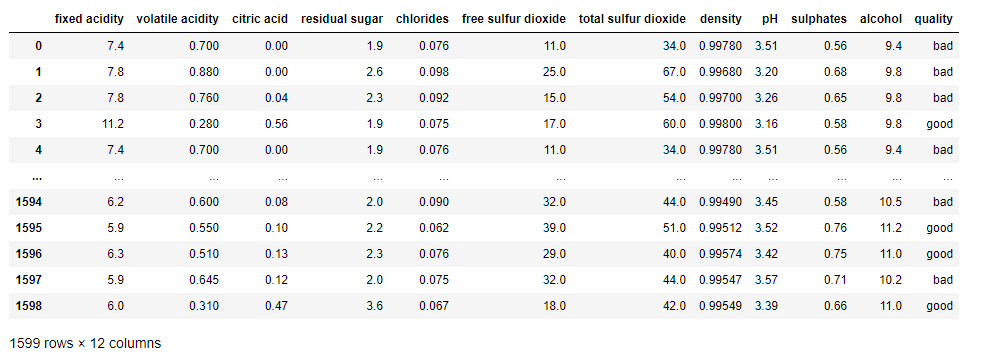
10 - sulfati

11 - alkohol

Klasa:

12-kvalitet (dobro ili loše)

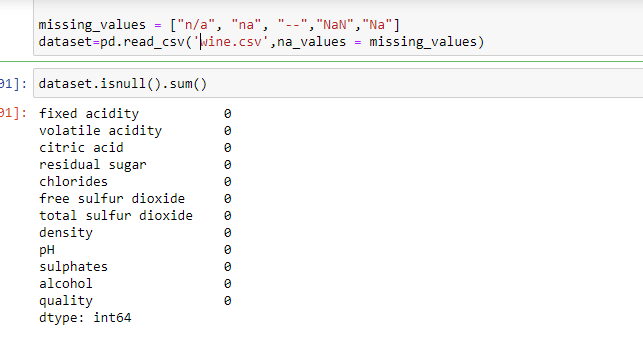
Deo skupa podataka prikazan je na sledećoj slici:



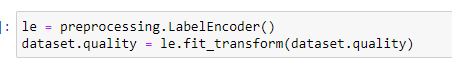
Opis projekta:

Pomoću funkcije read\_csv učitavamo skup podataka. Podaci su učitani u promenljivu dataset.

Pomoću funkcije dataset.isnull().sum() proverava se da li postoje podaci koji nedostaju. Ovom skupu podataka ne nedostaju podaci.



Za prevođenje atributa kategoričkog tipa u numerički, koristi se klasa LabelEncoder() i funkcija fit\_transform.

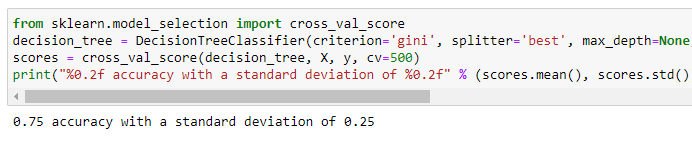


Promenljiva X će sadržati skup atributa na osnovu kojih treba odrediti klasu. Korišćenjem funkcije iloc, uzimaju se svi redovi i sve kolone iz dataseta, tj svi atributi, osim poslenjeg, koji je atribut klase (quality). Promenljiva y predstavlja klasu i njoj se dodeljuje vrednost 12. kolone iz dataseta.



Pošto nemamo dostupan poseban test skup, a želimo da ktvrdimo tačost klasifikatora upotrebićemo podatke koji su nam dostupni. Koristi se unakrsna validacija.

Procenjuje se tačnost klasifikatora (stabla odlučivanja), podelom skupa podataka,kreiranjem modela i izračunavanjem rezultata 500 uzastopnih puta za naš dataset koji sadrži 1599 instanci. Ovakav klasifikator bi dao tačnost od 75%, sa standardnom devijacijom koja iznosi 0,25.



Nadalje se skup podataka deli na trening set i test set korišćenjem funkcije train\_test\_split .

Nakon toga se kreira objekat klase DecisionTreeClassifier iz biblioteke scikit-learn.

Prametri kod kreranja objekta klase DecisionTreeClassifier su sledeći:

Criterion: “gini” - Funkcija za merenje kvaliteta podele. Gini predstavlja Ginijev indeks za meru nečistoće čvora. Može biti i entropy.

Splitter: “best” - Strategija koja se koristi za odabir podele na svakom čvoru. Koristi se strategija “best” za odabir najbolje podele.

max\_depth=None - Maksimalna dubina stabla. Ako je “None”, tada se čvorovi proširuju sve dok svi listovi ne postanu čisti ili dok svi listovi ne sadrže manje uzoraka od min\_samples\_split.

min\_samples\_split=2 - Minimalni broj uzoraka potreban za razdvajanje internog čvora.

min\_samples\_leaf=1 -Minimalan broj uzoraka potreban da bude na čvoru lista. Tačka razdvajanja na bilo kojoj dubini uzeće se u obzir samo ako ostane min\_samples\_leaf trening uzoraka u levoj i desnoj grani.

min\_weight\_fraction\_leaf=0 – Svi uzorci imaju jednaku težinu, jer je ovaj parametar postavljen na 0.

max\_features=None. Broj karakteristika koje treba uzeti u obzir prilikom traženja najbolje podele. Postavljen je na None, što znači da je max\_features=n\_features,tj ukupnom broju karakteristika.

random\_state=None- Kontroliše slučajnost procene. Karakteristike se uvek nasumično permutiraju pri svakoj podeli, čak i ako je spliter podešen na „best“. Kada je max\_features <n\_features, algoritam će nasumično odabrati max\_features pri svakoj podeli, pre nego što među njima pronađe najbolji split. Ali najbolje pronađeno razdvajanje može se razlikovati u različitim prolazima, čak i ako je max\_features = n\_features. U slučaju da je poboljšanje kriterijuma identično za nekoliko podela, jedna podela mora biti izabrana nasumično.

max\_leaf\_nodes=None. Neograničen je broj listova.

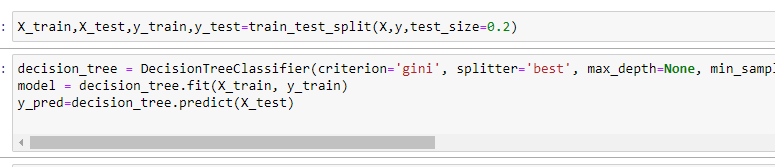
min\_impurity\_decrease=0.0- Čvor će se razdvojiti ako to razdvajanje indukuje smanjenje nečistoće veće ili jednako ovoj vrednosti.

min\_impurity\_split=None- Prag za rano zaustavljanje u formiranju stabla odluivanja. Čvor će se razdvojiti ako je njegova nečistoća iznad praga.

class\_weight=None- težina povezana sa klasom. Koristi se kod višestruke klasifikacije.

ccp\_alpha=0.0- Parametar složenosti koji se koristi za Minimal Cost-Complexity Pruning. Minimal Cost-Complexity Pruning je algoritam, koji se koristi za orezivanje drveta kako bi se izbegao over-fitting. Biće izabrano poddrvo sa najvećom složenošću troškova koja je manja od ccp\_alpha. Podrazumevano se ne vrši orezivanje (pruning).

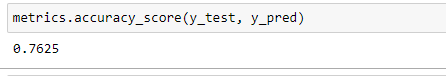
Korišćenjem funkcije fit, kreira se model koji se trenira, a zatim se za test skup podataka (X\_test), predvidja vrednost klase y (y\_pred), pomoću funkcije predict.



Performanse klasifikacije:

Performanse će biti određene korišćenjem funkcija iz sklearn.metrics.

Tačnost klasifikacije (accuracy) - Predstavlja broj slogova sa tačno dodeljenom klasom podeljen sa ukupnim brojem slogova. Tačnost je 76,25%.



U skupu trening slogova postoji P pozitivnih slogova i N negativnih slogova. Za svaki slog rezultat dobijen od klasifikatora se poredi sa poznatom klasom kojoj slog pripada.

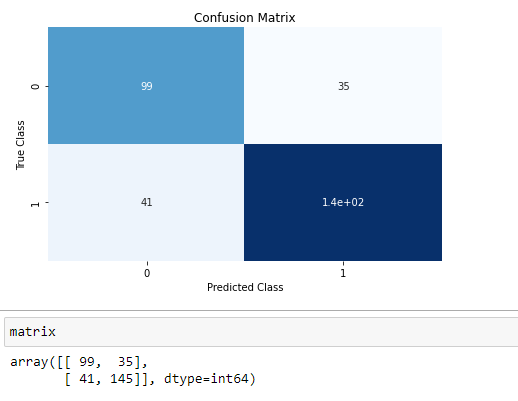
Mere: ◦ True positives (TP) – pozitivni slogovi koje je klasifikator ispravno prepoznao kao pozitivne

◦ True negatives (TN) – negativni slogovi koje je klasifikator ispravno prepoznao kao negativne

◦ False positives (FP) – negativni slogovi koje je klasifikator pogrešno prepoznao kao pozitivne

◦ False negativies (FN) - pozitivni slogovi koje je klasifikator pogrešno prepoznao kao negativne

Matrica konfuzije sadrži podatke o svim navedenim sogovima. Matrica konfuzije omogućava analizu tačnosti klasifikatora.



TP= 145

TN=99

FP=35

FN=41

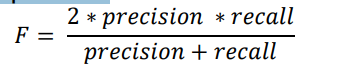
Ostale performanse možemo prikazati pooću funkcije classification\_report.

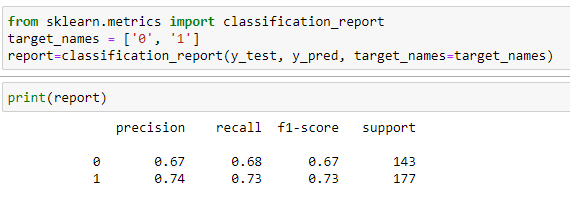
Preciznost (precision) – procenat pozitivno klasifikovanih slogova koji su zaista pozitivni . Za klasu 0 iznosi 67%, dok za klasu 1 iznosi 74%. Može se zaključiti da je klasifikator precizniji kod prepoznavanje klase 1.

Kompletnost (recall) – procenat ispravno klasifikovanih pozitivnih slogova u odnosu na ukupan broj pozitivnh slogova. Za klasu 0 iznosi 68%, dok za klasu 1 iznosi 73%.

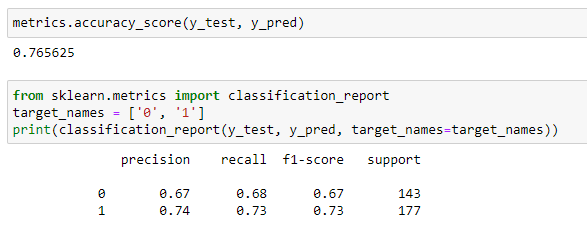
Dakle, klasifikator je tačnosti 76,25% , što ga čini dobrim klasifikatorom, ali je bolje preoznaje pozitivne slogove.

F mera (F measure, F1 score, F-score) - harmonic mean. Daje podjednak značaj preciznosti i kompletnosti.





Ukoliko se koristi entropy, dobijaju se malo bolji rezultati. Tačnost od 76,5 %. I u ovom slučaju klasifikator bolje prepoznaje pozitivne slogove. Zbog toga će se koristiti ovakav klasifikator.



Selekcija osobina:

Za selekciju osobina korišćena je eliminacija unazad - Backward elimination. Koraci:

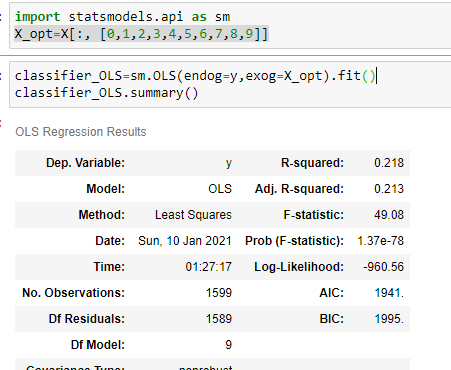
1. Izabere se takozvani nivo značajnosti. Najčešće je to 5% (0LS=0,05).

2. Počinje se sa celim skupom atributa (x1 , x2 ,..., xn ), formira se i testira klasifikator, zabeleže performanse.

3.Izabere se predkitor (atribut) koji ima najveću P-vrednost. Ukoliko je ta vrednost veća od 0LS, taj atribut (prediktor) se može ukloniti.

Ovo se ponavlja dok svi atributi ne budu imali P-vrednost manju od 0LS. Tada je model spreman, i može se vršiti predviđanje bez atributa koji su eliminisani. Novi modeli mogu dati bolje performanse.

Deo postupka eliminacije dat je na slici:



Odabrani su određeni atributi,a zatim prikazane P vrednosti i uklonjene one koje su veće od 0SL.

Na ovaj način se dobija manja tačnost kod klasifikacije, ali je znatno redukovana dimenzionalnost, radi se sa manje atributa.

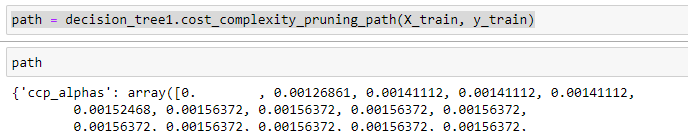
Pruning:

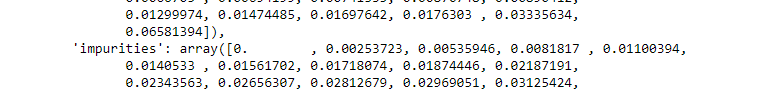
U post-pruning-u kreira se veliko stablo, a zatim se orezuje natrag kako bi se dobilo poddrvo takvo da dobijamo najnižu stopu grešaka u testiranju.

Problem ovog algoritma taj što ne želimo da idemo na svako podstablo i izaberemo svako od njih da bismo izračunali promenu u stopi greške.

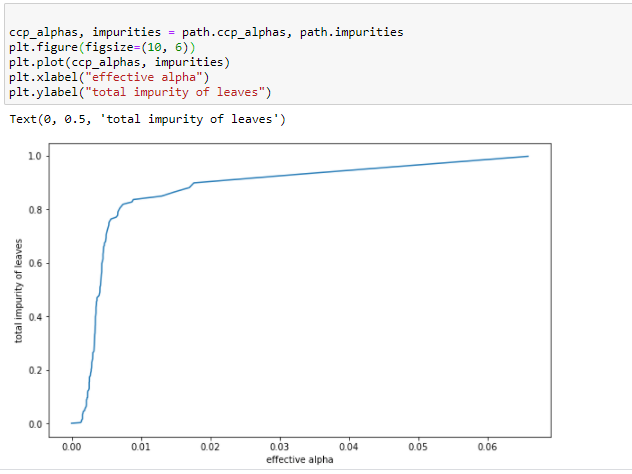
**Cost complexity Pruning** ili **Weakest link Pruning** nam pomažu u tome. Uvodi se termin alfa. Biramo samo ona stabla koja su indeksirana sa alfa.

Da bismo dobili različite primenljive vrednosti alfa i odgovarajuće nečistoće u čvorovima listova koji su uvedeni zbog toga, možemo da koristimo funkciju cost\_complexity\_pruning\_path prisutnu u klasi DecisionTreeClassifier. Tako ćemo dobiti vrednosti za nečistoće i parametar alfa.

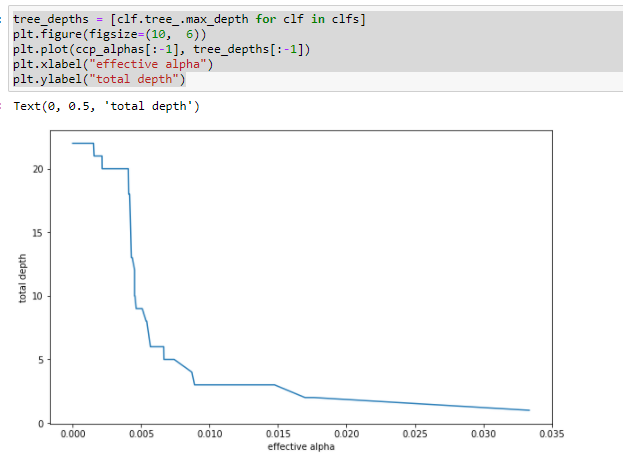


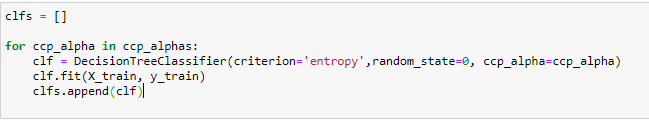


Ovo se može predstaviti i pomoću grafika:

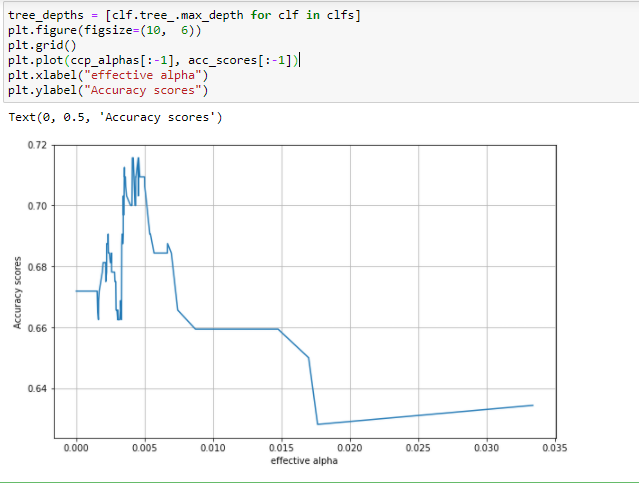


Postoji i jaka povezanost između dubine stabla i alfa. Ovo se može videti na grafiku:



Može se odrediti optimalan parametar alfa. Za različite vrednosti parametra alfa, kreira se stablo odluke i razmatra tačnost klasifikacije. 

Na sledećoj slici je prikazana tačnost u odnosu na odabranu vrednost za alfa.



Najveća tačnost se dobija za alfa koje je približno 0,005.