**3.Domaći zadatak – Naivni Bajes**

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.

Za predviđanje u mašinskom učenju možemo iskoristiti i verovatnoću. Metoda **Naivni Bajes** je jedna od najčešće korišćenih probabilističkih metoda za svrhu višeklasne klasifikacije. Jednostavna je za razumevanje, a pored toga postiže i dobre rezultate u širokom spektru problema.

Poreklo imena:

**Naivna** - smatra se da uticaj vrednosti jednog atributa na klasnu pripadnost ne zavisi od vrednosti drugih atributa

**Bajesova** – bazira se na Bajesovoj formuli za izračunavanje uslovne verovatnoće

*P(A|B)* - predstavlja verovatnoću da se desio događaj A pod uslovom da se desio događaj B

*P(A)* - verovatnoća da se desio događaj A

**Gausov model Naivne Bajesove metode**:

*P(C|F1, F2, F3 ,…, Fn)* - predstavlja verovatnoću da objekat predstavljen vektorom (F1, F2, F3 ,…, Fn) pripada klasi C

*P(C)* – predstavlja verovatnoću pojavljivanja klase C u trening skupu

*P(F1, F2, F3 ,…, Fn|C)* – predstavlja verovatnoću da entitet koji ima vrednosti atributa (F1, F2, ... , Fn) pripada klasi C

*P(F1, F2, F3 ,…, Fn)* – predstavlja verovatnoću da entitet ima vrednosti atributa (F1, F2, ... , Fn)

ParametarP(C=c) se određuje na osnovu trening skupa kao:

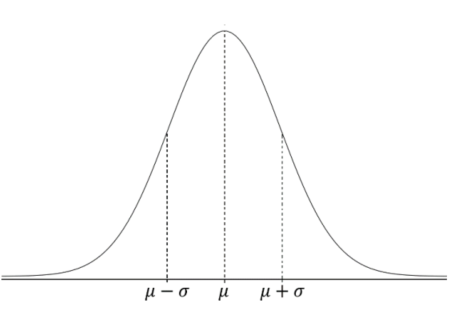
||Dc|| - predstavlja broj objekata klase C  
||D|| - predstavlja ukupan broj trening objekata

ParametarP(Fi = fi, C = c) se određuje na osnovu trening skupa kao:

µc(Fi) - predstavlja srednju vrednost atributa Fi u objektima klase C

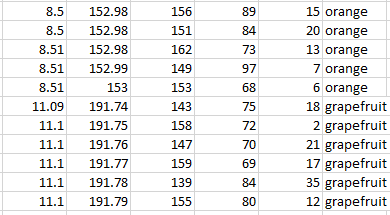
c(Fi) – predstavlja standardnu devijaciju atributa Fi u objektima klase C

Pri čemu funkcija f(x, µ, ) predstavlja izraz:



**Dataset**:

Za potrebe domaćeg zadatka korišćen je „citrus.csv“ skup podataka (u .csv formatu), preuzet sa sajta <https://www.kaggle.com/>. Na sledećoj slici priložen je odlomak iz dataseta, na kom su prikazane karakteristike voća, klasifokovane u klase „orange“ i „grapefut“.



Skup podataka sadrži 10.000 zapažanja. Takođe sadrži 4 atributa i jedan atribut koji će predstavljati klasu.

Atributi:

1 – prečnik (cm)

2 – težina (g)

3 – crvena boja (0-255)

4 – zelena boja (0-255)

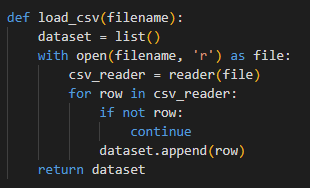
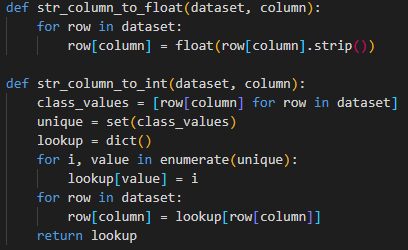
5 – plava boja (0-255)

Klasa:

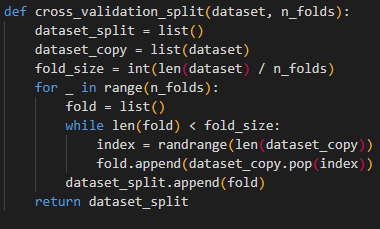
6 – „narandža“ ili „grejpfrut“

**Opis projekta:**

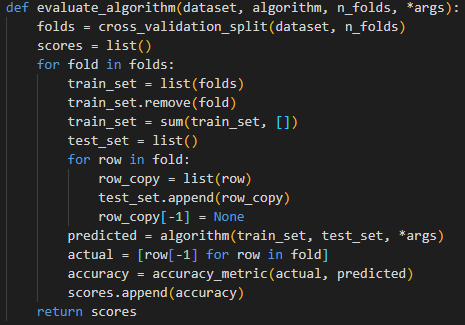
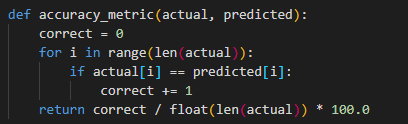
Prvi korak je učitavanje skupa podataka i pretvaranje učitanih podataka u brojeve koje možemo koristiti sa proračunima srednje i standardne devijacije. Za ovo koristimo pomoćnu funkciju **load\_csv()** za učitavanje podataka, **str\_column\_to\_float()** za pretvaranje string brojeva u float i **str\_column\_to\_int ()** za pretvaranje klasne kolone u celobrojne vrednosti.

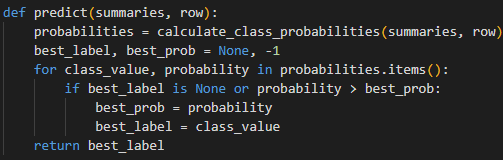
Sam algoritam ćemo proceniti pomoću k-fold unakrsne provere sa 10 foldova. To znači da će 10000/10 = 1000 zapisa biti u svakom foldu. Funkcija **cross\_validation\_split()** i njena implementacija prikazane su na slici ispod:



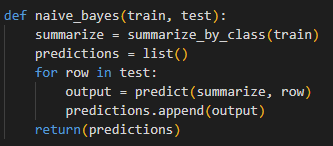
Korišćenje su pomoćne funkcije **evaluate\_algorithm()** za procenu algoritma sa unakrsnom validacijom i **accuracy\_metric()** za izračunavanje tačnosti predviđanja.

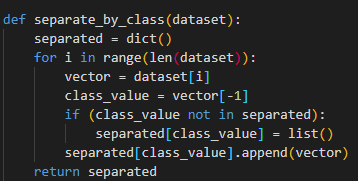
Funkcija nazvana **predict()** je implementirana za upravljanje proračunom verovatnoće novog reda koji pripada svakoj klasi i odabirom klase sa najvećom vrednošću verovatnoće.



Funkcija pod nazivom **naive\_baies()** je razvijena za upravljanje primenom Naive Baies metode (opisane na početku dokumenta), prvo učeći statistiku iz skupa trening podataka i koristeći ih za predviđanje skupa test podataka.

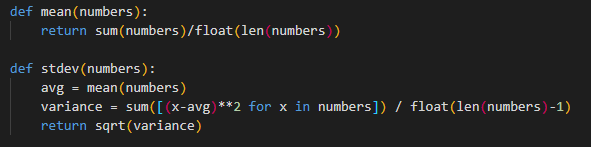
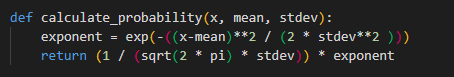


S obzirom da treba da izračunamo verovatnoću podataka prema klasi kojoj pripadaju, takozvanoj osnovnoj stopi, to znači da ćemo prvo morati da razdvojimo podatke o obuci po klasama. Relativno direktna operacija. Možemo kreirati dictionary gde je svaki key vrednost same klase, a zatim dodati listu svih zapisa kao value u rečniku. Ispod je funkcija nazvana **separat\_bi\_class ()** koja primenjuje ovaj pristup. Pretpostavlja se da je poslednja kolona u svakom redu vrednost klase.



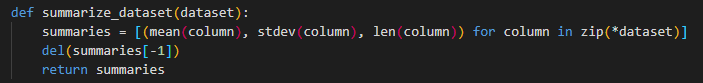
Za potrebe izračunavanja vrednosti funkcije

Kreirane su klase **mean(numbers)** – za izračunavanje srednje vrednosti neke karakteristike, **stdev(numbers)** – za izračunavanje standardne devijacije, **calculate\_probability(x, mean, stdev)** koja računa izraz prikazan iznad.

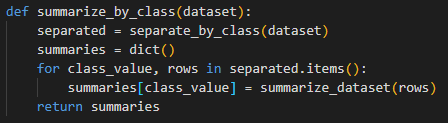
 

Izračunavanje statistika srednje vrednosti i standardne devijacije za svaki ulazni atribut ili svaku kolonu naših podataka možemo učiniti posredstvom skupljanjem svih vrednosti za svaku kolonu u listi i izračunavanjem srednje vrednosti i standardne devijacije na toj listi.

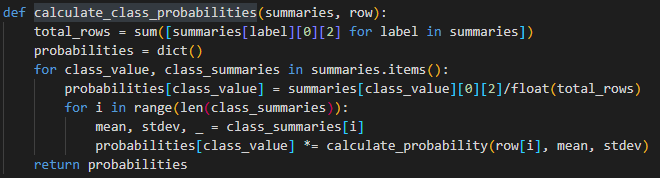
Ispod je funkcija nazvana **summarize\_dataset()** koja primenjuje ovaj pristup. Upotreba zip () funkcije će agregirati elemente iz svakog navedenog argumenta. Skup podataka prenosimo na funkciju zip () sa operatorom \* koji razdvaja skup podataka u posebne liste za svaki red. Funkcija zip () se zatim prevlači preko svakog elementa svakog reda i vraća kolonu iz skupa podataka kao listu brojeva. Zatim izračunavamo srednju vrednost, standardnu devijaciju i broj redova u svakoj koloni.



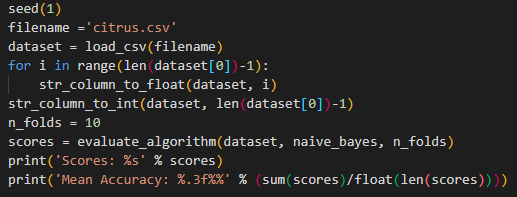
Iznad u dokumentu smo opicali funkciju **separat\_bi\_class()** za odvajanje skupa podataka u redove prema klasi, kao i funkciju **summarize\_dataset()** za izračunavanje zbirnih statistika za svaku kolonu. Sve ovo možemo složiti i rezimirati kolone u skupu podataka organizovane po vrednostima klase. Ispod je funkcija nazvana **summarize\_bi\_class()** koja implementira ovu operaciju. Skup podataka se prvo deli prema klasi, a zatim se statistika izračunava za svaki podskup. Rezultati u obliku spiska statističkih podataka se zatim čuvaju u dictionary prema vrednosti klase.



Verovatnoće se izračunavaju za svaku ulaznu vrednost u redu pomoću Gaussove funkcije gustine verovatnoće i statistike za tu kolonu i te klase. Verovatnoće se množe zajedno kako se akumuliraju. Ovaj postupak se ponavlja za svaku klasu u skupu podataka. Na kraju se vraća dictionary verovatnoća sa po jednom stavkom za svaku klasu.



**Performanse:**



Pokretanje programa ispisuje srednje ocene tačnosti klasifikacije na svakom prelazu unakrsne provere, kao i srednju ocenu tačnosti. Možemo videti da je srednja tačnost oko 92%.

