|  |
| --- |
| Fakultet tehničkih nauka |
| **Klasifikacija karcinoma** |
|  |

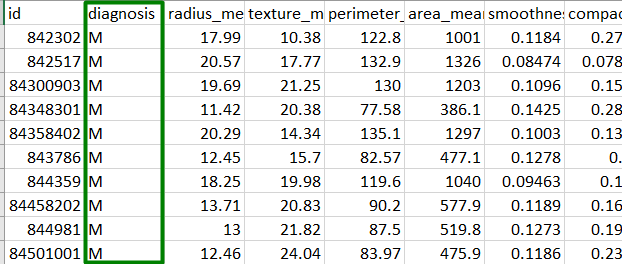
|  |
| --- |
| RA 197/2021 - Dušanić Andrej  Maj 2024. |

**Uvodna reč**

Na osnovu dobijenih podataka potrebno je formirati model koji će uspešno razlikovati beningni od malignog karcinoma.

**Obrada (priprema) podataka**

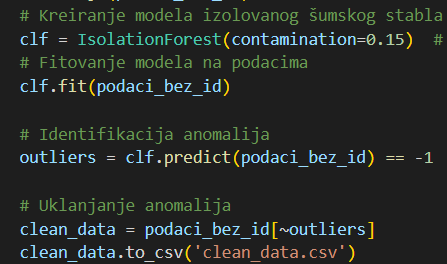
Da bismo uopšte krenuli sa izradom projekta neophodno je razumevanje podataka. Podaci su dati u csv fajlu „Cancer\_data.csv“. To radimo uvidom u csv fajl, kao i pozivom funkcija „**head**“ i „**info**“.



S obzirom da je zadatak da obucimo program da vrši klasifikaciju karcinoma očigledno je da nam je kolona od interesa, kolona „diagnosis“. Isto tako je uočljivo da kolona „id“ nema nikakvog uticaja na kolonu od interesa tako da vršimo **izbacivanje kolone „id“**, a s obzirom da su u koloni „diagnosis“ vrednosti stringovi, **vršimo konverziju: M->1, B->0**. Uvidom u podatke primetna je i kolona **„Unnamed:32“** koja ima sve NaN vrednosti i koju **izbacujemo.**

Podatke još testiramo provjerom da li ima uzoroka koji su duplikati, ili da li postoje NaN vrednosti, s obzirom da nema prelazimo u sledeću fazu projekta.

* Anomalije (outlieri),   
  Podaci koji značajno odstupaju od ostalih posmatranja u skupu podataka. Oni mogu značajno uticati na performanse i tačnost modela mašinskog učenja, pa je važno pravilno ih identifikovati i tretirati. U ovom slučaju ćemo ih izbaciti.



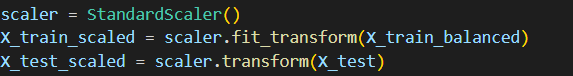
Balansiranje  
se radi kada su klase neuravnotežene u skupu podataka. Kao što vidimo u našem skupu podataka ima znatno više 1 od 0. Ovo može dovesti do toga da model bude pristrasan ka češćim klasama, što rezultira lošom tačnošću za manje zastupljene klase.  


Problem balansiranja sam rešio upotrebom **SMOTE** tehnike iz biblioteke imblearn. Ona sintetički generiše nove uzorke iz manjinske klase. Cilj SMOTE-a je da se poveća broj uzoraka manjinske klase kako bi se smanjila nebalansiranost između klasa, što može poboljšati performanse modela mašinskog učenja.



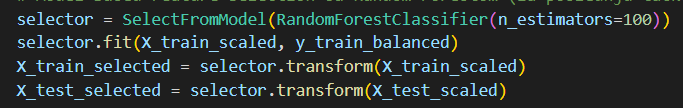


* Skaliranje podataka  
  Transformacija podataka kako bi se različite karakteristike dovele na istu skalu. Ovo je posebno važno u algoritmima mašinskog učenja koji se oslanjaju na udaljenosti između podataka, kao što su: K Nearest Neighbors (KNN), Support Vector Machines (SVM), i Gradient Descent bazirani algoritmi (npr. linearna i logistička regresija).



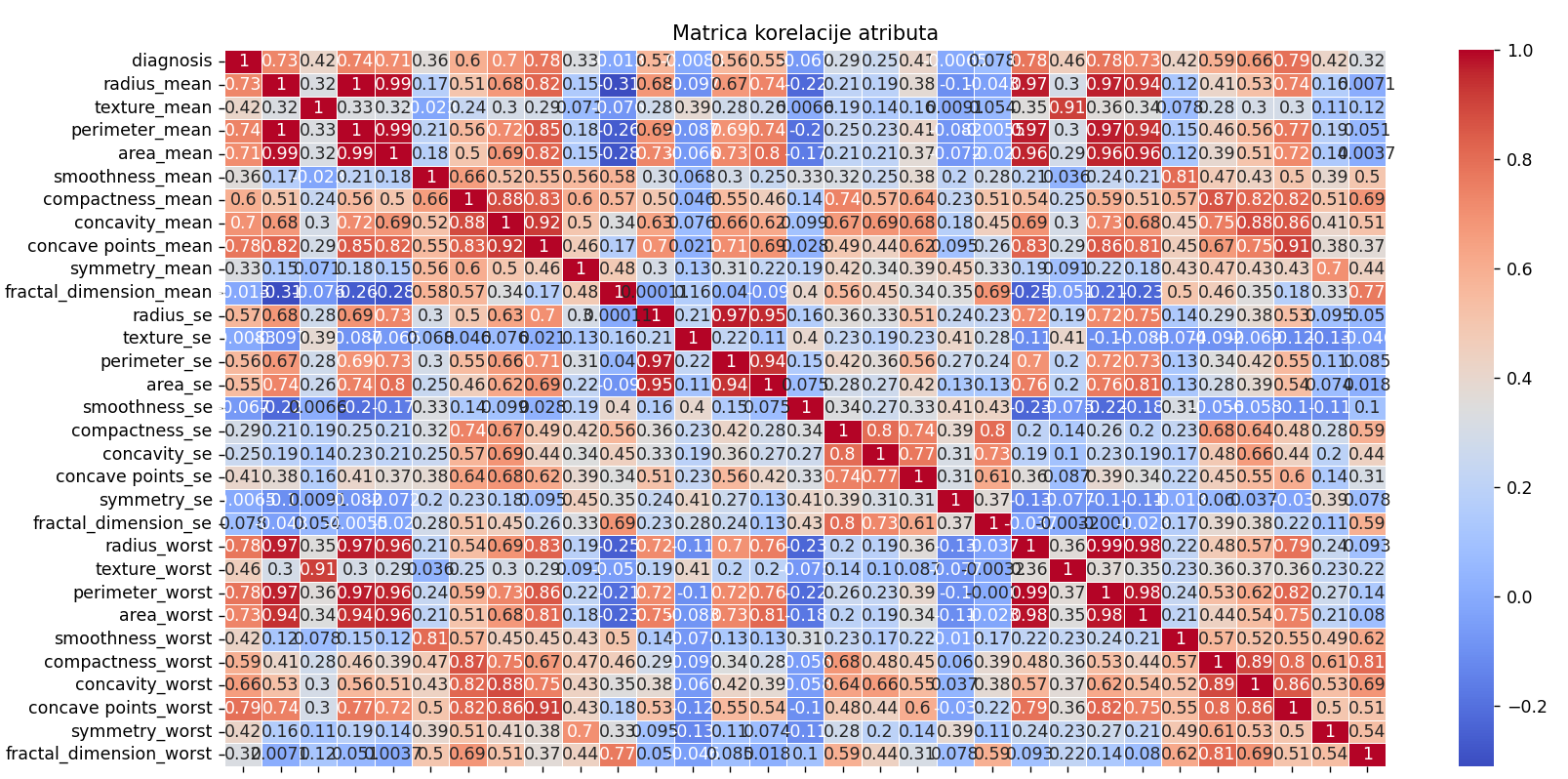
* Selektovanje Najbitnijih Atributa (Feature Selection)

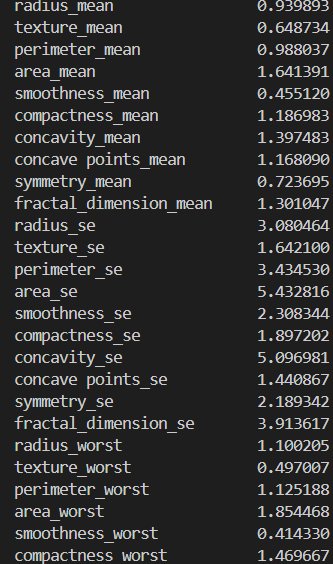
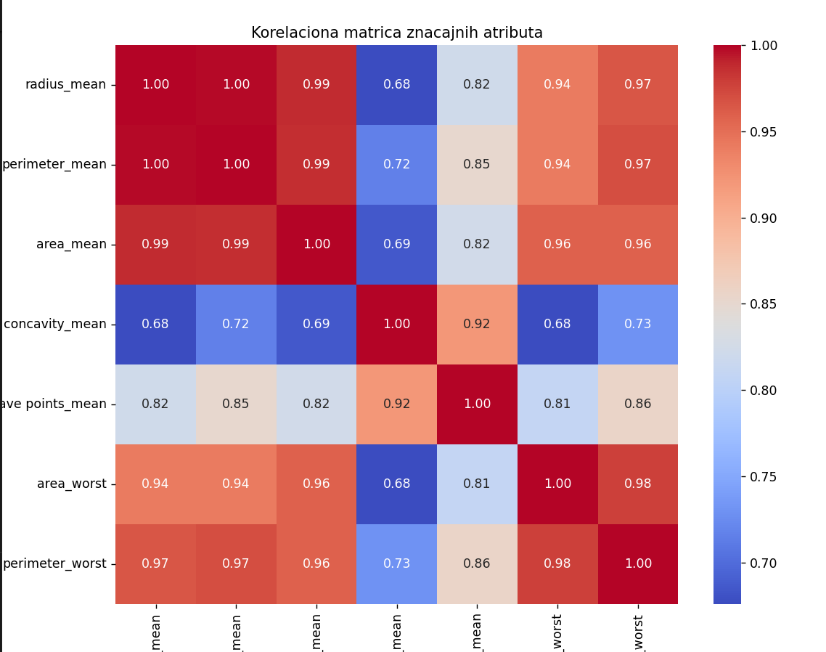
Selektovanje najbitnijih atributa je proces izbora podskupa relevantnih karakteristika (atributa) iz skupa podataka, koji su najvažniji za treniranje modela. Ovaj proces je ključan za poboljšanje performansi modela, smanjenje pretreniravanja (overfitting), i smanjenje vremena treniranja modela.



**Eksplorativna analiza skupa**

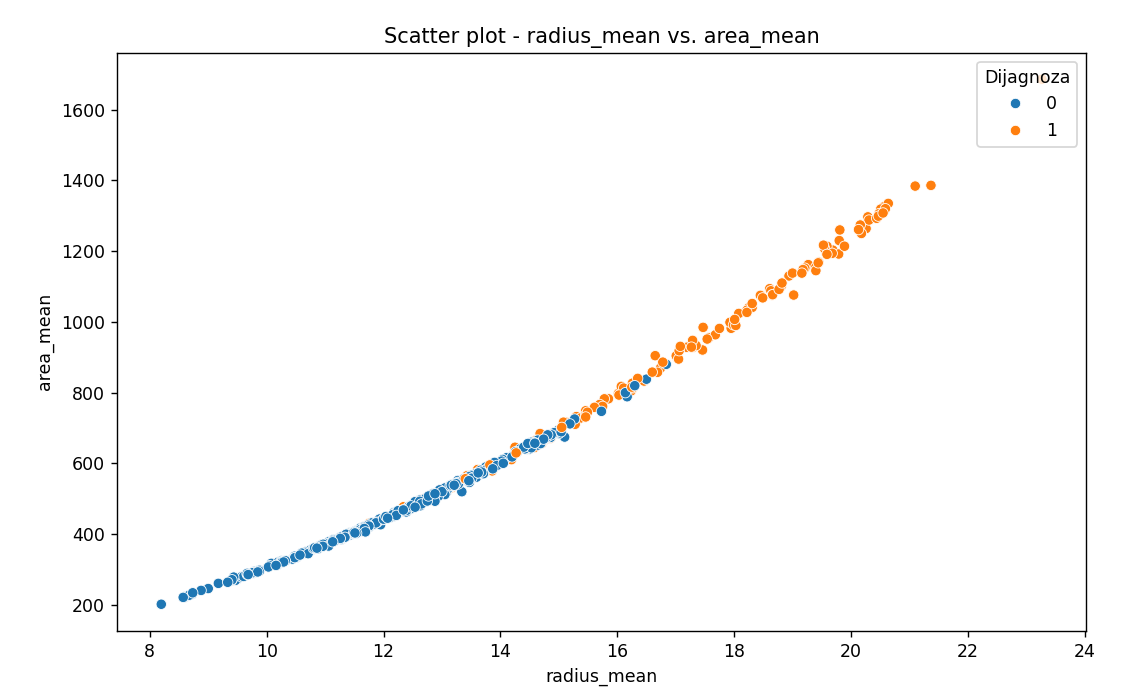
* Korelaciona matrica  
  Korelaciona matrica nam prikazuje koeficijente korelacije između više promenljivih. Korelacija meri snagu i pravac linearne veze između dve promenljive. Ona se koristi u eksplorativnoj analizi podataka **govori nam o odnosu između različitih atributa** u skupu podataka.   
  S obzirom da je broj atributa ogroman i sama preglednost korelacione matrice je loša tako da sam uz pomoć funkcije „**skew**“ odredio atribute koji su uticajniji (veći broj) i formirao korelacionu matricu samo za neke određene atribute.  
  U nastavku je korelaciona matrica svih i odabranih atributa.





Vrednosti u svakoj korelacionioj matrici su između -1 i 1. Što je vrednost u matrici bliža 1---pozitivna linearna korelacija (kako se jedna promenljiva povećava, druga se takođe povećava). Što je negativnija--- negativna linearna korelacija (kako se jedna promenljiva povećava, druga se smanjuje). Za vrednosti blizu 0 ne postoji linearna korelacija. Korelacija iznad 0.7 ili ispod -0.7 se često smatra visokom i može ukazivati na to da se jedna promenljiva može predvideti na osnovu druge.

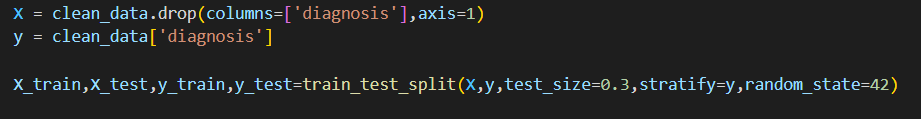
* Raspodela ciljnog atributa u odnosu na neke nezavisne promenljive  
  Ovaj postupak nam pomaže da bolje razumemo odnos između karakteristika i ciljne promenljive, što je od suštinske važnosti za izgradnju efikasnih modela mašinskog učenja.

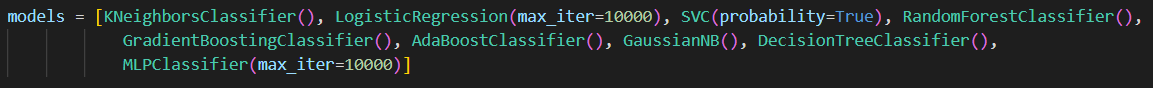


U kodu sam iskoristio: **Line plot, Scatter plot, violine plot, pie plot, Boxplot, Bar plot i Histogram.**

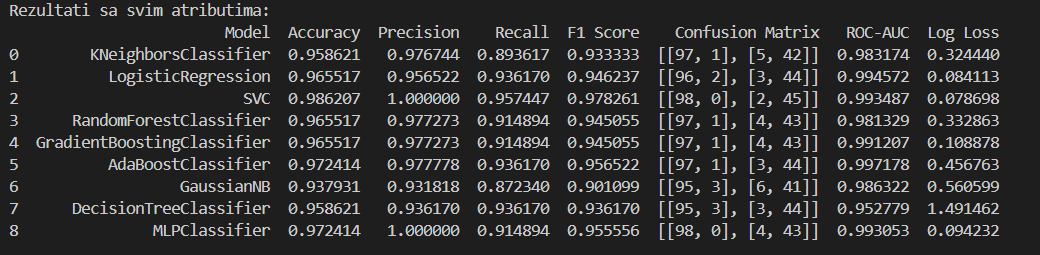
**Odabir pogodnog algoritma**

* Zavisi od: Performansi algoritma, Kompleksnosti algoritma, Veličine skupa podataka, Dimenzionalnosti skupa podataka
* Često se problem odabira pogodnog algoritma rešava tako što se kreiraju „bazični“ modeli, proveri njihova uspešnost, pa potom radi sa onim koji od starta daje povoljne rezultate



Modeli koje sam koristio su:

**Obučavanje i testiranje modela zasnovanog na odabranom algoritmu, Analiza dobijenih rezultata**

Uporedni rezultati (svi atributi):

Ovoga vidimo da su svi upotrebljeni modeli dobri, tačnost se u zavisnosti od upotrebljenog modela kreće preko 93%.   
Isto tako analizu rezultata možemo posmatrati i preko drugih metrika. Metrike koje sam upotrebio:

--Precision (Preciznost):

Preciznost predstavlja proporciju tačno pozitivnih instanci među svim instancama koje su model označile kao pozitivne.

Formula: True Positives / (True Positives + False Positives)

Preciznost je korisna kada su lažno pozitivne klase skuplje od lažno negativnih, tj. kada je važno minimizovati broj lažno pozitivnih instanci.

--Recall (Odziv):

Odziv predstavlja proporciju tačno pozitivnih instanci među svim stvarno pozitivnim instancama u testnom skupu.

Formula: True Positives / (True Positives + False Negatives)

Odziv je koristan kada su lažno negativne klase skuplje od lažno pozitivnih, tj. kada je važno minimizovati broj lažno negativnih instanci.

--F1 Score:

F1 skor je harmonijski srednja vrednost preciznosti i odziva. Predstavlja balans između preciznosti i odziva.

Formula: 2 \* (Precision \* Recall) / (Precision + Recall)

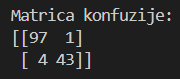
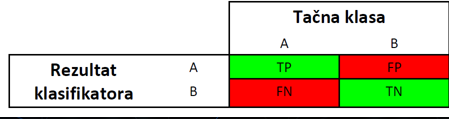
--ROC-AUC:

Površina ispod ROC krive (ROC-AUC) meri sposobnost modela da razlikuje između klasa. Viša vrednost označava bolje performanse modela.ROC kriva (Receiver Operating Characteristic) predstavlja grafik zavisnosti između stope lažno pozitivnih i stope istinito pozitivnih klasifikacija modela pri različitim pragovima klasifikacije. Grafik ROC krive sam takođe implementirao u kodu.

--Log Loss:

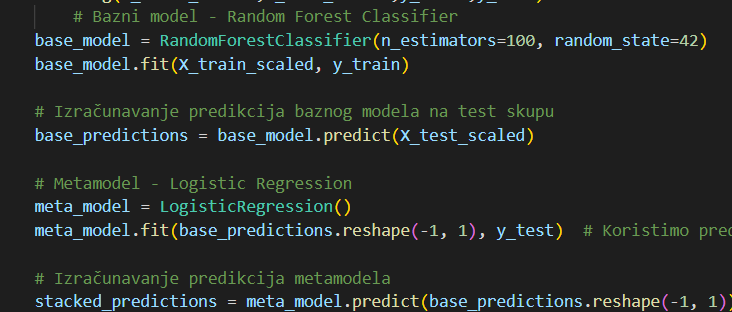
Log Loss meri performanse verovatnoća predviđanja klase. Niži log gubitak ukazuje na bolje performanse modela.Log Loss se koristi kada je potrebno meriti nesigurnost predikcije, posebno kod verovatnoćnih modela kao što su logistička regresija ili modeli koji daju verovatnoću klase.

Matrica konfuzije

Tabela koja se koristi u problemima klasifikacije kako bi se prikazala performansa modela na testnom skupu podataka, posebno u kontekstu klasifikacije binarnih problema (kada postoji samo dve klase). Ova matrica prikazuje broj tačno pozitivnih (TP), tačno negativnih (TN), lažno pozitivnih (FP) i lažno negativnih (FN) klasifikacija modela. Evo kako izgleda matrica konfuzije: 

* Stacking:

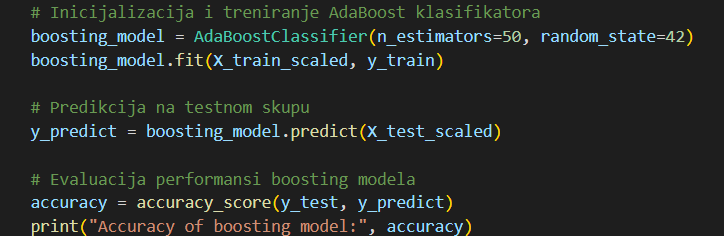
Tehnika ansambla u mašinskom učenju koja kombinuje više modela kako bi postigla bolje performanse predikcije nego što bi to postigao pojedinačni model. Ova tehnika je jedna od vrsta ansambl metoda, koje kombinuju predikcije više osnovnih modela kako bi se dobila finalna predikcija. Faze:

1. Osnovni modeli: Osnovni modeli se treniraju na originalnom skupu podataka.
2. Metamodel: Predikcije osnovnog modela se koriste kao ulazne značajke za metamodel (obično jednostavan model poput logističke regresije ili linearnog SVM-a).
3. Konačna predikcija: Metamodel se koristi za generisanje finalnih predikcija na testnom skupu podataka.

* Boosting:

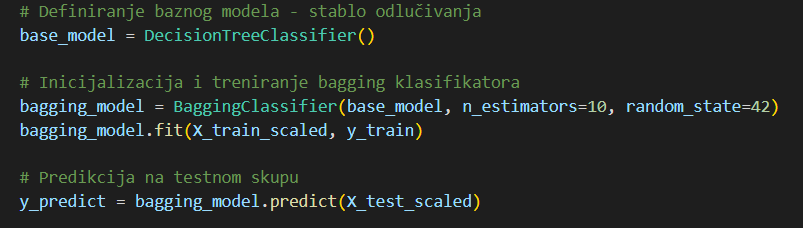
ansambl tehnika u mašinskom učenju koja kombinuje više slabijih modela kako bi se stvorio jači model. Osnovna ideja iza boostinga je da se uzastopno trenira niz slabih modela, a svaki sledeći model se fokusira na greške koje su napravljene od strane prethodnih modela. Na kraju, svi ovi modeli se kombinuju kako bi se stvorio jači model koji daje bolje performanse u odnosu na pojedinačne modele. Faze:

1. Inicijalizacija težina: Instancama u skupu podataka se dodeljuju početne težine.
2. Treniranje modela: Prvi model se trenira na originalnom skupu podataka. Zatim se ocenjuje performansa tog modela na skupu podataka, pri čemu se greške računaju za svaku instancu.
3. Ažuriranje težina: Težine instanci se ažuriraju kako bi se više pažnje posvetilo instancama koje su pogrešno klasifikovane.
4. Treniranje sledećeg modela: Sledeći model se trenira na skupu podataka sa ažuriranim težinama. Ovaj proces se ponavlja dok se ne dostigne zadovoljavajuća performansa ili se dostigne unapred definisani broj modela.
5. Kombinacija modela: Svi modeli se kombinuju u finalni model, često korišćenjem težinskih suma.



* Baging:

Ansambl tehnika koja se koristi u mašinskom učenju za poboljšanje stabilnosti i smanjenje varijabilnosti modela.

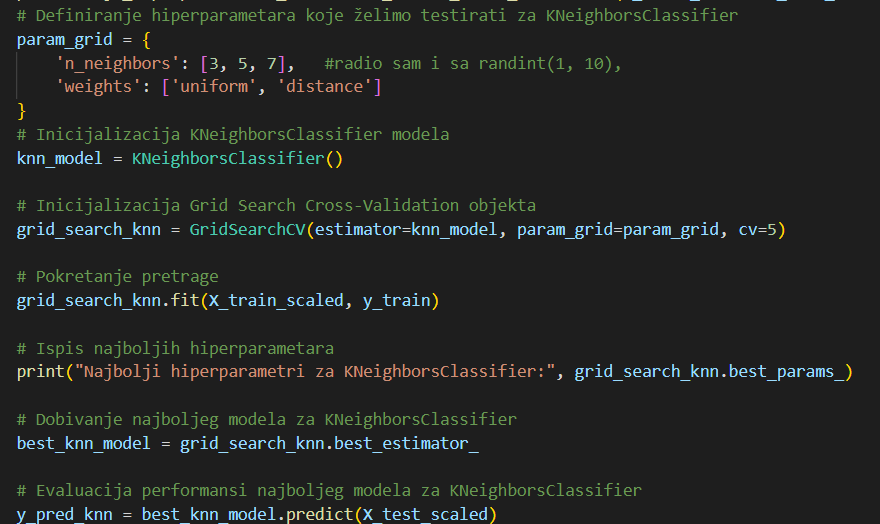
Osnovna ideja bagginga je da se stvori više varijanti istog osnovnog modela, ali treniranih na različitim podskupovima trening podataka. Ovi podskupovi se često uzimaju slučajno sa ponavljanjem (bootstrap uzorci). Zatim se svaki od ovih modela kombinuje kako bi se dobila konačna predikcija. Primeri bagging algoritama uključuju Random Forest, BaggingClassifier u scikit-learn biblioteci za Python, i Bagged Trees u XGBoost biblioteci. Ja sam koristio **BaggingClassifier** :

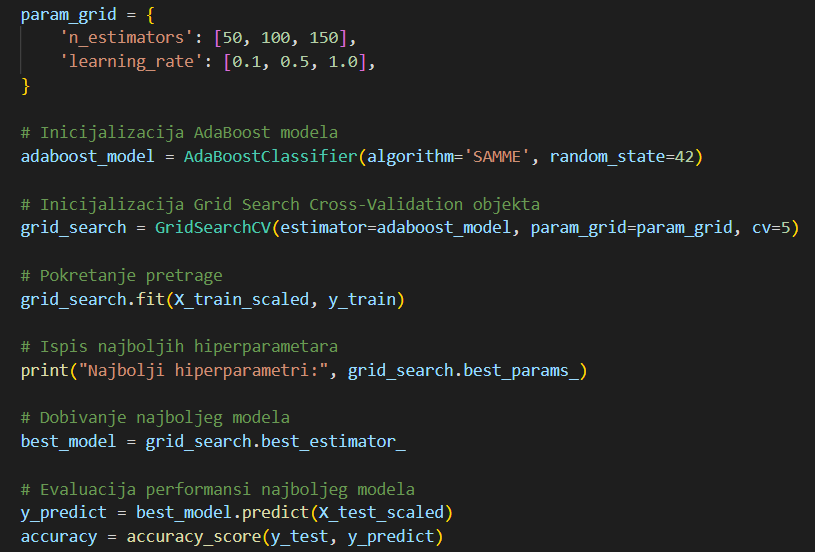
* Podešavanje hiperparametara:

Proces pronalaženja optimalnih vrednosti hiperparametara za model mašinskog učenja. Hiperparametri su parametri koji se ne uče tokom treninga modela, već se postavljaju pre samog treninga i mogu značajno uticati na performanse modela.

Postoji nekoliko tehnika za podešavanje hiperparametara, uključujući: Grid Search Cross-Validation, Randomized Search Cross-Validation, Bayesian Optimization, Genetski algoritmi

Ja sam koristio: **Grid Search Cross-Validation** i **Randomized Search Cross-Validation.**

****

****

* Unakrsna validacija:

Tehnika evaluacije performansi modela mašinskog učenja koja se koristi kako bi se procenila generalizacija modela na novim, neviđenim podacima. Ova tehnika podeljuje skup podataka na nekoliko podskupova, gde se model trenira na nekim od preklopa i evaluira na preostalim podacima. Ovaj postupak se ponavlja više puta, svaki put sa drugačijom podelom podataka, i na kraju se prosečna performansa modela uzima kao konačna mera.

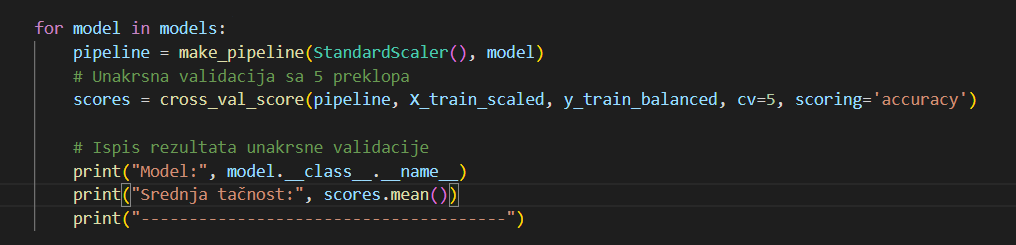
Postoji nekoliko vrsta unakrsne validacije:K-struka unakrsna validacija,

Ponovljena K-struka unakrsna validacija, Liven-out unakrsna validacija,

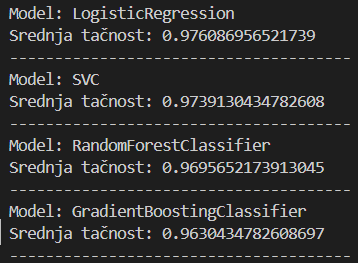
Grupna unakrsna validacija.

Unakrsna validacija je važna tehnika u mašinskom učenju jer omogućava realniju procenu performansi modela i pomaže u sprečavanju prenaučenosti (overfitting) ili podnaučenosti (underfitting).

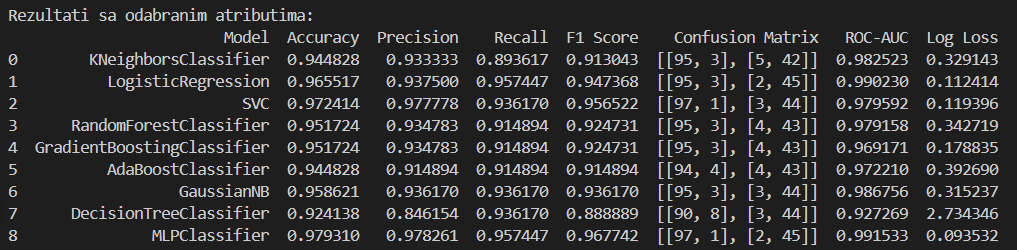
Ja sam koristio unakrsnu validaciju sa 5 preklopa:



Nakon čega smo za svaki model dobili srednju tačnost:

****

Pozivom samo najbitnijih atributa rezultati se blago menjaju, podaci su prostiji, program je samim tim brži.

****

Zaključak

Na kraju možemo da zaključimo da nam je generalno poznavanje problema i podataka nad kojim radimo ipak ključan i najkompleksniji korak. Odabir modela nad ovim problemom i podacima nije toliko bio ključan jer su sve tačnosti do negde zadovoljavajuće, međutim očigledno je da DecisionTreeClassifier daje najlošije rezultate u odnosu na ostale korištene algoritme (srednja tačnost mu je za 3 lošija od prvog narednog).