PROCJENA KVALITETE VINA

Orić, Mihaela

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

PROCJENA KVALITETE VINA

Završni rad

Mihaela Orić

Osijek, 2021.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 16.09.2021.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime studenta:	Mihaela Orić
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R4252, 26.07.2018.
OIB studenta:	67864991307
Mentor:	Izv. prof. dr. sc. Emmanuel Karlo Nyarko
Sumentor:	dr.sc. Ivana Hartmann-Tolić
Sumentor iz tvrtke:	
Naslov završnog rada:	Procjena kvalitete vina
Znanstvena grana rada:	**Umjetna inteligencija (zn. polje računarstvo)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	16.09.2021.
Datum potvrde ocjene Odbora:	22.09.2021.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 27.09.2021.

Ime i prezime studenta:	Mihaela Orić
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R4252, 26.07.2018.
Turnitin podudaranje [%]:	2

Ovom izjavom izjavljujem da je rad pod nazivom**: Procjena kvalitete vina**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Emmanuel Karlo Nyarko

i sumentora dr.sc. Ivana Hartmann-Tolić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. Uvod

Vino je alkoholno piće koje se najčešće konzumira diljem svijeta. Njegova kvaliteta je bitna kako kod potrošača, tako i kod proizvođača jer je konkurencija u proizvodnji velika. Najstariji način provjere kvalitete vina je test kušanjem. Ako vino ne zadovolji test kušanja, cijeli se postupak ponavlja ispočetka uz mnoge preinake što oduzima mnogo vremena i novca.

Kupovina vina može biti veliki izazov ako je cilj iskušati nešto novo. Veća cijena ne garantira da je vino kvalitetnije. Što je onda pokazatelj dobroga vina? Deklaracija boce može sadržavati cijeli kemijski sastav, ali on kupcu u većini slučajeva ne pomaže. Što kada bi postojao način da se na temelju osnovnih kemijskih i fizikalnih obilježja određenog vina dobije njegova kvaliteta na ljestvici od 1 do 10 koju svatko može razumjeti? U ovom radu pokušat će se riješiti taj problem metodama strojnoga učenja. Strojno učenje je područje u računarstvu koje je najbliže vezano za umjetnu inteligenciju i statistiku. Koriste se algoritmi za predviđanje i učenje obrazaca kako bi se automatiziralo donošenje odluka.

U drugom poglavlju predstavit će se pregled područja koji opisuje radove slične ovome i koji su bili osnova za pisanje ovog rada.

Klasifikacija, u strojnom učenju, ima kao cilj odrediti klasu ili kategoriju, odnosno razvrstati podatke na temelju zadanih kriterija. Ona će biti upotrijebljena u ovom radu pomoću programskog jezika Python u okruženju Visual Studio Code, uz biblioteke Scikit-Learn, Numpy, Panda koje su objašnjene u trećem poglavlju ovog rada. U istom tom poglavlju opisani su i modeli i metode potrebne za provedbu klasifikacije.

Za učenje i testiranje upotrijebit će se online baza podataka crvenih i bijelih vina koja je predstavljena u četvrtom poglavlju. Podaci dohvaćeni iz Sveučilišta u Londonu bit će podijeljeni i obrađeni te pripremljeni za modele za strojno učenje.

Ta obrada i podjela podataka te primjena modela odrađena je u petom poglavlju. Nakon primjene prikazani su i evaluirani rezultati.

U zadnjem poglavlju ovog rada iznesen je zaključak.

1.1 Zadatak završnog rada

Kupovina novih neiskušanih vina može biti dobar ili loš pogodak. Na temelju skupa podataka dobivenog fizikalno-kemijskim mjerenjima treba izraditi klasifikacijski odnosno regresijski model za procjenu kvalitete vina.

2. Pregled područja

Ovim problemom predviđanja vrste i kvalitete vina pomoću strojnog učenja bave se autori u radu [1] i taj rad je glavna osnova ovom radu. U tome radu autori koriste istu baza vina kao i u ovom i na toj bazi vrši se strojno učenje. Određena su tri regresijska modela od kojih je najuspješniji bio model temeljen na stroju s potpornim vektorima koji će se odrediti i u ovome radu.

Četvero brazilskih znanstvenika u svome su radu [2] također primijenili strojno učenje za klasifikaciju vina. Iako se u njemu ne klasificira vino po kvaliteti ili vrsti, nego po zemlji porijekla, taj rad je sličan ovome jer koristi dva algoritma za klasifikaciju koja su primijenjena u ovom radu. Autori navedenog rada opisali su metodologiju koja može biti primijenjena i u drugim proizvodima osim vina kako bi se prepoznala svojstva koja najviše utječu na klasifikaciju, smanjio broj dimenzija podataka i najvažnije, povećala uspješnost klasifikacije.

Na isti način se prepoznaju najvažnija svojstva vina pri klasifikaciji i u radu [3]. Rezultati autora ovog rada dokazuju da se preciznost modela znatno povećava ako se za učenje koriste samo svojstva koja znatno pridonose kvaliteti umjesto učenja sa svim svojstvima.

Rad [4] također se bavi predviđanjem kvalitete vina, ali pristupa svojstvima koja utječu na kvalitetu na drugačiji način. Za razliku od ostalih navedenih primjera klasifikacije, ovdje model ne predviđa kvalitetu ovisno o svojstvima vina, nego ovisno o svojstvima grožđa od kojeg se vino proizvodi. Baza podataka o grožđu nastajala je u Grčkoj dvije godine.

U radu [5] primjenjuju se tehnike strojnog učenja i inteligentna analiza podataka bez korištenja skupih kemijskih testova da bi se dobila detaljna baza podataka za učenje. Ovaj rad je dokaz da čak i manje detaljna svojstva mogu biti pogodna kao ulazne varijable modela za učenje. Cilj naučenoga modela nije predvidjeti kvalitetu, nego prepoznati je li poznato talijansko vino sorte Nebbiolo autentično ili lažno. Krivotvorenje ove sorte vina laganjem o korištenoj sorti grožđa ili o njegovom geografskom porijeklu prouzrokovalo je štetu od nekoliko milijuna eura što je pokazatelj koliko je ovakva tema rada važna.

3. Primijenjeni modeli i biblioteke

Porastom primjena strojnog učenje raste i popularnost programskog jezika Python jer je upravo on najpogodniji za rješavanje svih problema koje donosi strojno učenje. Jednostavnost sintakse i čitljivost koda pomažu programerima da svoju pažnju usmjere na to što pišu, umjesto na to kako će to napisati. Uz to, Python nudi veliki sustav biblioteka koje omogućavaju lako pristupanje i manipuliranje podacima i bazama [6].

U ovom radu, za klasifikaciju vina korištena su dva algoritma, algoritam slučajnih šuma i algoritam stroja s potpornim vektorima. S obzirom da oba klasifikatora rade s binarnim klasama, odnosno izlazi su im dvije klase, potrebno je koristiti malo drugačiji pristup kod procjenjivanja kvalitete.

Za procjenjivanje kvalitete koristit će se jedan nasuprot ostalih (engl. *One-vs-Rest* - OVR) klasifikator. OVR je klasifikator kod ne-binarnih klasa, kada postoji više od dva izlaza i on omogućuje da se tih više klasa klasificira na isti način kao da su samo dvije klase. Bez obzire koliko klasa se klasificira, OVR klasifikator uvijek radi binarno. Svaka klasa se uspoređuje sa svim preostalim klasama tako da je usporedba binarna. OVR funkcionira na sljedeći način:

1) Uspoređivanje izlaza 0 s [izlazom 1 i izlazom 2]
2) Uspoređivanje izlaza 1 s [izlazom 0 i izlazom 2]
3) Uspoređivanje izlaza 2 s [izlazom 0 i izlazom 1]

Za ispis rezultata, odnosno preciznosti modela na testnome skupu koristit će se klasifikacijski izvještaj (engl. *Classification report*) i matrica zbunjenosti (engl. *Confusion matrix*).

Jedan način za provjeru preciznosti je matrica zbunjenosti. Ona pokazuje koliko je 0 model identificirao ispravno kao 0 ili krivo kao 1 i koliko je 1 model identificirao ispravno kao 1 i lažno kao 0.

Tablica 3.1. Legenda matrice zbunjenosti

	Predviđena 0	Predviđena 1
Stvarna 0	Pravi negativ	Lažni pozitiv
Stvarna 1	Lažni negativ	Pravi pozitiv

Drugi način za provjeru preciznosti je pomoću klasifikacijskog izvještaja koji daje povratne informacije o preciznosti naučenog modela. On koristi prave i lažne negative te prave i lažne pozitive iz matrice zbunjenosti. U izvještaju te vrijednosti nisu iskazane u broju pozitiva i negativa nego u indeksu preciznosti (engl. *Precision*). Ta preciznost označava točnost pretpostavljenih pozitiva, a dobiva se iz sljedeće formule:

$$preciznost = pravi\ pozitivi\ /\ (pravi\ pozitivi + lažni\ pozitivi) \quad (3.1).$$

Osim preciznosti, izvještaj donosi i odziv (engl. *Recall*) koji pokazuje koliko je pozitiva ispravno prepoznato, a dobiva se iz formule [15]:

$$odziv = pravi\ pozitivi\ /\ (pravi\ pozitivi + lažni\ negativi) \quad (3.2).$$

3.1. Algoritam slučajnih šuma

Algoritam slučajnih šuma (engl. *Random forest classifier*) je fleksibilan algoritam za strojno učenje koji je jednostavan za korištenje. Pruža precizne rezultate većinu vremena čak i bez pretjeranog namještanja parametara. Jedan je od najčešće korištenih algoritama zbog svoje jednostavnosti i svestranosti. Svestran je jer se može koristiti za klasifikaciju i regresiju. U njegovom nazivu nalazi se riječ šuma (engl. *Forest*), a zove se šumom jer se sastoji od „upakiranih" stabala odluke (engl. *decision tree*).

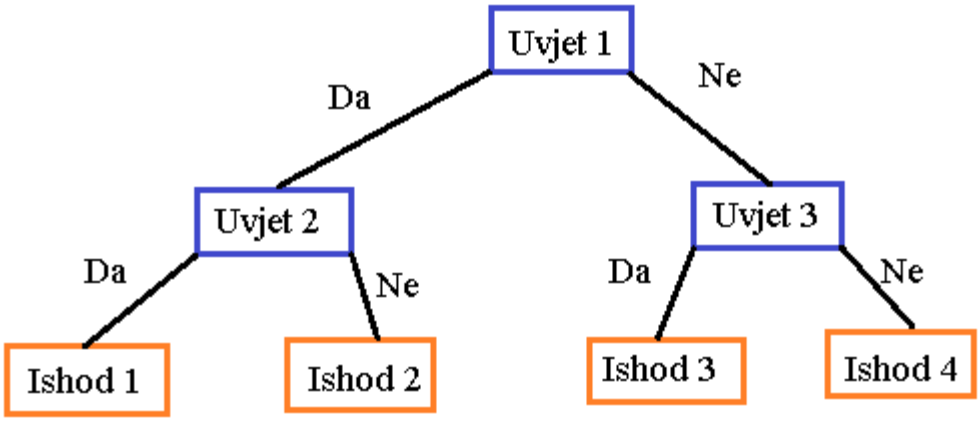
Stablo odluke je vrsta strojnog učenja koje je temeljeno na binarnom stablo. Ono spada u nenadzirani način strojnog učenja i korišten je za regresiju i klasifikaciju.

Svaka grana stabla prikazuje ishod odnosno vrijednost koju čvor može poprimiti. Odlučivanje koja od dvije vrijednosti će biti odabrana vrši se pomoću skupa pravila ako-onda (engl. *if-then*). Ovisno o parametrima, podaci se dijele i tako se putuje kroz stablo sve dok se ne dođe do dna stabla. Stablo odluke moguće je izraditi na nekoliko načina: ID3 algoritam, C4.5 i CART (engl. *Classification and regression trees*) algoritam [17]. Kroz stablo se prolazi od gore prema dolje i ono je izrađeno rekurzivno.

Što je više stabala za odlučivanje, to je algoritam precizniji. Metoda „pakiranja" (engl. *bagging*) označava kombiniranje različitih modela strojnog učenja s ciljem postizanja preciznijih rezultata. Ako se radi o klasifikaciji, tada algoritam slučajne šume pravi više različitih stabala koja glasaju, a u slučaju regresije pronalaze srednju vrijednost.

Iako povećanjem broja stabala se povećava i preciznost algoritma, ako se stablo pretrpa performanse se smanjuju. Pretrpavanje stabala nastupa kada se doda previše grana, odnosno previše mogućnosti odabira.

U algoritmu slučajne šume se pri uzorkovanju zanemari 33% podataka. Ti podaci nazivaju se „podaci izvan torbe" ili (engl. *out of bag* - OOB) i određeni su nasumično tako što svako stablo nasumično bira koje podatke će koristiti. Procjena greške tog skupa podataka je usporediva s procjenom greške cijele slučajne šume. To znači da se praćenjem OOB skupa može odrediti koliko daleko će trening podataka ići. Stabiliziranje OOB greške označava kraj treninga [16].

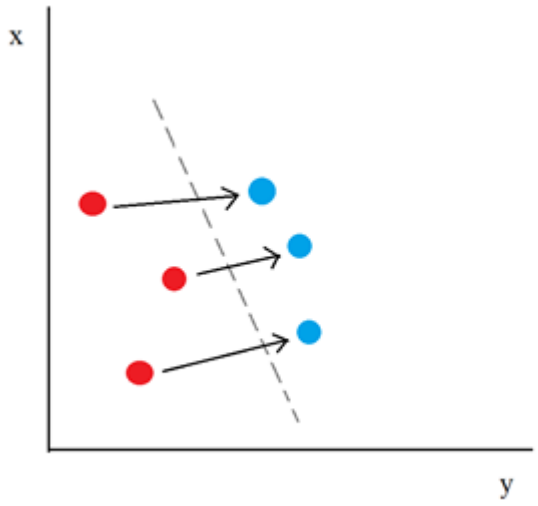


Slika 3.1. Jednostavan primjer stabla odluke.

3.2. Algoritam stroja s potpornim vektorima

Algoritam stroja s potpornim vektorima ili (engl. *Support Vector Machines* - SVM) jedan je od najpopularnijih i najsnažnijih algoritama za strojno učenje koji postoji, a nastao je 1990-ih. Može se koristiti za klasifikaciju i regresiju, ali većinom se koristi kao klasifikator. U SVM algoritmu, svaki podatak prikazuje se kao točka u n-dimenzionalnom prostoru, a broj dimenzija određen je svojstvima koje imamo. U ovom slučaju, to bi bilo 11 osi u sustavu, a za svaki uzorak vina, vrijednosti obilježja bi bile vrijednosti na tim osima. Klasifikacija se odvija tako što se pronalazi hiperravnina koja razdvaja te dvije klase što bolje. Dimenzije hiperravnine ovise o broju svojstava seta podataka. Ako sustav ima dva svojstva, hiperravnina će biti pravac. Ako se radi o tri svojstva, hiperravnina će biti dvodimenzionalna ploha. U ovom radu koristi se 11 obilježja, što znači da je hiperravnina 10-dimenzionalna ploha. U nazivu ovog klasifikatora stoje potporni vektori, a oni su svoj naziv dobili jer su ti vektori najbliži hiperravnini i oni određuju poziciju te ravnine.

Zadataka SVM algoritma je pronaći granicu (ili granice) između svih točaka u sustavu na način da su na različitim stranama hiperravnine vrijednosti svojstava koje spadaju u različite kategorije. Algoritam nastoji maksimizirati tu ravninu tako da je njena svaka točka na ravnini (na svakom od vektora) najbliža vrijednosti drugog uzorka. [14]



Slika 3.2. Prikaz potpornih vektora.

Kada se koristi za regresiju, naziva se još i regresor potpornih vektora (engl. *Support Vector Regressor* - SVR). Tada se bazira na regresijskog metodi osnovanoj na jezgrama koja uzima nelinearne podatke u stvarnom prostoru i stavlja ih u više dimenzija koristeći jezgrine funkcije. Neke od tih jezgri su linearne, sigmoidne, radijalne i polinomne.

3.3. Python biblioteke

Kako bi se navedeni algoritmi mogli lako primijeniti na podatke, koristi se nekoliko Python biblioteka koje su opisane u nastavku.

Pandas je snažan, brz, fleksibilan i besplatan alat za analizu i obradu otvorenog koda. Osnovni cilj mu je biti osnova svake praktične analize podataka u Pythonu. Izveden je kao *DataFrame* objekt s ugrađenim indeksiranjem, odnosno kao raspodijeljena kolekcija podataka organiziranih u imenovane stupce. Sadrži alate za pisanje i čitanje u različitim strukturama podataka i formatima, a neki od tih formata su (engl. *Comma-Separated Values* - CSV) i tekstualne datoteke, SQL baze podataka i Microsoft Excel datoteke. Nudi jednostavno dodavanje ili brisanje redaka ili stupaca i preoblikovanje i transformiranje setova podataka. Osim dijeljenja tih podataka na manje skupove, omogućuje i njihovo spajanje [7].

Seaborn je biblioteka osnovana na matplotlib biblioteci i pruža sučelje za crtanje informativnih i atraktivnih statističkih grafova što pomaže u boljem razumijevanju podataka. Ova biblioteka omogućuje korisniku da se usmjeri na proučavanje vizualiziranih podataka i na donošenje zaključaka na osnovu danih grafova umjesto trošenja vremena na njihovo crtanje [8].

Numpy (skraćeno od Numerical Python) je brza i svestrana biblioteka koja se često koristi u bilo kojem području znanosti ili inženjerstva. Osnova je rada s brojčanim podacima u Pythonu i koristi se u gore uvezenim bibliotekama [9].

Sci-kit learn (sklearn) je jedna od najkorisnijih biblioteka za strojno učenje u Pythonu. Sadrži mnogo učinkovitih alata za strojno učenje i modeliranje poput klasifikacije, regresije, grupiranja i smanjenje dimenzionalnosti, alati za modificiranje podataka. U njoj su sadržane mnoge tehnike nadziranog i nenadziranog učenja te standardni skupovi podataka. Sučelje sci-kit learn je jako jednostavno za korištenje [10].

Biblioteka sci-kit learn uvedena je pod nazivom „sklearn" a uz nju su zatim dodani razni dodaci koje ona pruža, poput „svm" za algoritam stroja s potpornim vektorima, „metrics" za prikaz povratnih informacija o preciznosti naučenog modela, „preprocessing" za obradu podataka prije samog učenja (za standardizaciju i normalizaciju), „model_selection" za podjelu podataka na set za učenje i testni set te „multiclass" za klasifikator za klasifikaciju ne-binarnih klasa.

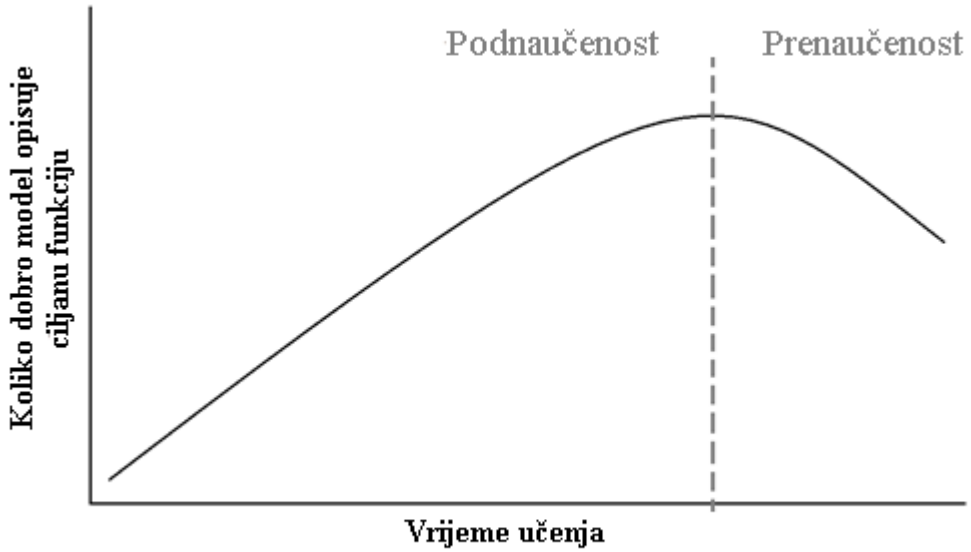
3.4. Podjela i obrada podataka

Osim što gore navedene biblioteke omogućavaju primjenu klasifikatora, također pružaju i alate za podjelu podataka koja je neizostavan dio strojnog učenja.

Podaci se dijele na dvije grupe: skup za učenje (engl. *train set*) na kojem se vrši učenje (za najbolju preciznost, skup za učenje trebao bi sadržavati većinu seta podataka, čak 80%) i skup za testiranje (engl. *test set)* koji služi za provjeru uspješnosti učenja (on je manji i sadrži 20% podataka).

Glavni razlog podjele podataka na ovakve grupe je da bi se izbjegla prenaučenost (engl. *Overfitting*), ali i podnaučenost (engl. *Underfitting*) jer oni uzrokuju lošu učinkovitost kod strojnog učenja. Statističko podudaranje (engl. *Statistical fit*) odnosi se na to koliko je dobro aproksimirana ciljana funkcija.
Prenaučenost se odnosi na model koji se pretjerano prilagođava skupu za učenje te on savršeno opisuje podatke. Tada, taj model uči sve pojedinosti skupa za učenje do te mjere da krene smanjivati učinkovitost modela na novom skupu, odnosno skupu za testiranje. Kod ovakvog slučaja, model je u obzir uzeo sve iznimke u skupu podataka i njih gledao kao pravilo, što znači da u testnom slučaju neće prepoznati koja pojedina obilježja su iznimke, nego će ih suditi prema pravilima za ostala obilježja. Do prenaučenosti često dolazi kod nelinearnih modela.
Podnaučenost se odnosi na model koji ne modelira ni skup za učenje niti se prilagođava novom skup. On ima lošu uspješnost na skupu za učenje te je logično da će njegova uspješnost biti loša i na skupu za testiranje. O njemu ne postoji previše diskusija niti objašnjenja kako ga riješiti jer se vrlo lako detektira i ispravi.

Da bi statističko podudaranje bilo što bolje, nastoji se pronaći sredina između prenaučenosti i podnaučenosti. Kada bi se vizualizirala promjena statističkog podudaranja obzirom na to kako traje proces učenja, na početku bi točnost skupa za učenje i skup za testiranje bila niska, ali bi se s vremenom povećavala. Tada je model u stanju podnaučenosti. Nakon nekog vremena, doći do prekretnice. Točnost skupa za učenje će sve više rasti, ali točnost skupa za testiranje krenut će padati. Taj trenutak označuje pojavu prenaučenosti. Savršeno statističko podudaranje je neposredno prije pojave prenaučenosti [12].



Slika 3.3. Ovisnost preciznosti modela i vremena učenja.

Slika 3.3. pokazuje koliko dobro naučeni model opisuje ciljanu funkciju ovisno o vremenu učenja modela. Isprekidana linija između podnaučenosti i prenaučenosti označava trenutak kada se događa najoptimalnije statističko podudaranje.

Nakon podjele podataka u skupove za učenje i testiranje, slijedi njihova obrada i transformacija.

Podaci prije obrade mogu biti u raznim oblicima. U ovom slučaju postoji jedanaest svojstava vina i skoro svako od tih svojstava izraženo je u različitim jedinicama. Tako postoji vrijednost limunskih kiselina koja će imati vrlo male vrijednosti (sve vrijednosti su manje od jedan), pH razine koje imaju vrijednosti od jedan do sedam, ali i ukupni sumpor-dioksidi koji su u nekim slučajevima veći i od pedeset. Model koji uči na podacima koji se ovoliko razlikuju u veličinama bio bi jako neprecizan. Razlika u dioksidima kada se uspoređuje 50 i 10 je velika i naspram nje razlika limunske kiseline između 0.10 i 0.20 jako je mala, čak i zanemariva, ali je jednako bitna. Korištenjem ovih veličina kakve jesu, svojstvo koje je većeg iznosa će zasjeniti svojstvo manjeg iznosa u odlučivanju. Varijable koje se mjere na različitim ljestvicama ne pridonose jednako analizi.

Dvije najpoznatije metode skaliranja podataka su normalizacija i standardizacija.

Normalizacija označava skaliranje realnih podataka u raspon između 0 i 1. Kod strojnog učenje, ne zahtjeva svaki skup podataka normalizaciju. Ona se koristi samo kada se svojstva nalaze u različitim rasponima.

Standardizacija označava pomicanje distribucije svakog atributa na način da se postigne srednja vrijednost jednaka 0 i standardna devijacija jednaka 1. Nakon standardizacije moguće je uspoređivati varijance pojedinih atributa. Način na koji se ovo postiže:

$$standardizirani\ set\ podataka = \frac{(skup\ podataka - srednja\ vrijednost(skup\ podataka)}{standardna\ devijacija\ (skup\ podataka)} \quad (3.3).$$

Teško je znati treba li podatke normalizirati ili standardizirati prije primjene modela. Postoji pravilo da je normalizacija dobar odabir transformiranja podataka kada se zna da podaci nisu distribuirani u obliku zvona (Gaussova krivulja) [13]. Normalizacija se također koristi kada se žele transformirati podaci uz zadržavanje nekih različitosti u ljestvicama. Standardizacija pretpostavlja da je distribucija podataka oblika Gaussove krivulje kao što je slučaj kod linearne regresije, logističke regresije i linearne diskriminantne analize. Korisna je i kada se atribute želi uspoređivati (preko standardne devijacije).

U ovom radu standardizirani su skupovi za učenje i testiranje. Bitno je uočiti da se standardiziraju samo ulazni podaci. Izlazni podaci ostaju kakvi jesu jer za svaki model postoji samo jedan izlaz i on treba ostati kakav je. Kvaliteta kao izlaz ostaje na ljestvici od 1 do 10, a vrsta kao izlaz ostaje 0 ili 1, odnosno crno ili bijelo vino.

Metoda primijenjena na skupu za učenje da bi se taj skup skalirao je *fit_transform()*.Ta metoda upotrijebljena je i da bi se zapamtili parametri skaliranja tog skupa. Ovu metodu moguće je podijeliti na dva dijela: *fit* i *transform*. Prva se poziva metoda *fit()* koja računa srednju vrijednost i varijancu svakog svojstva prisutnog u skupu podataka. Zatim se poziva funkcija *transform()* koja je objašnjena u nastavku.

Transform() metoda koristi srednju vrijednost i varijancu naučenu iz skupa za učenje i koristi te parametre kod transformiranja. Primjenjuje se na svakom svojstvu skupa za učenje i skupa za testiranje.

Kada bi se na oba skupa podataka koristila metoda *fit_transform(),* tada bi se u oba slučaja kreirala nova srednja vrijednost i varijanca i model bi učio i na testnom skupu. U tom slučaju model ne bi pokazivao stvarne rezultate učenja jer bi novi parametri utjecali na rezultate modela. Model je naučen na skupu za učenje i isti taj model mora biti primijenjen i na testnom skupu jer inače to ne bi bio testni skup nego novi skup za učenje [13].

Tablica 3.2. Podaci prije standardizacije.

	Fiksna kiselost	Hlapljiva kiselost	Limunske kiseline	Sulfati	Alkohol
1	6.60e+00	2.40e-01	3.50e-01	3.70e-01	1.05e+01
2	8.30e+00	2.80e-01	4.80e-01	6.20e-01	1.24e+01
3	7.70e+00	7.15e-01	1.00e-02	5.70e-01	1.18e+01

Tablica 3.3. Podaci poslije standardizacije.

	Fiksna kiselost	Hlapljiva kiselost	Limunske kiseline	Sulfati	Alkohol
1	-0.46822	-0.59951	0.20547	-1.08327	0.01377
2	0.85149	-0.35285	1.10406	0.59858	1.60872
3	0.38571	2.32961	-2.14471	0.26221	1.10505

4. Baza s crnim i bijelim vinima

Vinski stručnjaci kažu da vina razlikujemo ovisno o njihovom okusu, mirisu i boji.

Dva skupa podataka na kojima je ovaj rad zasnovan nastala su na uzorcima vina iz sjevernog Portugala i dohvaćena su iz repozitorija Sveučilišta u Londonu („UCL Machine Learning Repository“) [11]. Koristit će se baza podataka s obilježjima i kvalitetama crnih vina i baza s istim obilježjima bijelih vina. Oba skupa podataka sadrže jedanaest fizikalnih i kemijskih svojstava, a to su: fiksna kiselost (ukupna kiselost – hlapljiva kiselost), hlapljiva kiselost (koliko kiselosti nestane hlapljenjem vina, plinovite kiseline), ukupni sumpor-dioksid, kloridi (količina soli u vinu), pH razina (u vinu pH razina označava kiselost), slobodni sumpor-dioksid (štiti vino blagim kvarenjem mikroba), gustoća, neprevreli šećer (koliko šećera je ostalo nakon fermentacije), limunska kiselina (slaba organska kiselina koja je prirodno dodana u voće), sulfati (dodani sulfati koji čuvaju svježinu vina i štite ga od oksidacije i bakterija) i alkohol (izražen u postotku). Posljednje obilježje je kvaliteta koju su odredili ispitivači testom kušanja. Svako vino ocijenili su na ljestvici od nula (najlošija kvaliteta) do deset (najbolja kvaliteta).

Također je dodano još jedno obilježje koje označava vrstu vina (je li vino bijelo ili crno).

Pomoću funkcija iz biblioteke Pandas moguće je proučiti strukturu podataka.

Tablica 4.1. Prvih pet vina iz baze podataka.

Fiksna kiselost	Hlapljiva kiselost	Limunske kiseline	Neprevreli šećer	Kloridi	Slobodni sumpor-dioksid	Ukupni sumpor-dioksid	Gustoća	pH	Sulfati	Alkohol	Kvaliteta	Vrsta vina
7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.998	3.51	0.56	9.4	5	0
7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.996	3.20	0.68	9.8	5	0
7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.997	3.26	0.65	9.8	5	0
11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.990	3.16	0.58	9.8	6	0
7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.997	3.51	0.56	9.4	5	0

Da bi se lakše kontroliralo jesu li podaci pravilno raspoređeni, postoje funkcije u biblioteci Pandas koje govore koliko vina ima svake kvalitete, odnosno svake vrste.

Tablica 4.2. Raspored vina po kvalitetama

Kvaliteta	Broj vina
6	2836
5	2138
7	1079
4	216
8	193
3	30
9	5

Tablica 4.3. Raspored vina po vrsti

Vrsta vina	Broj vina
0	4898
1	1599

Pozivom ove funkcije lako se utvrđuje da baza sadrži 4898 crvenih i 1599 bijelih vina.

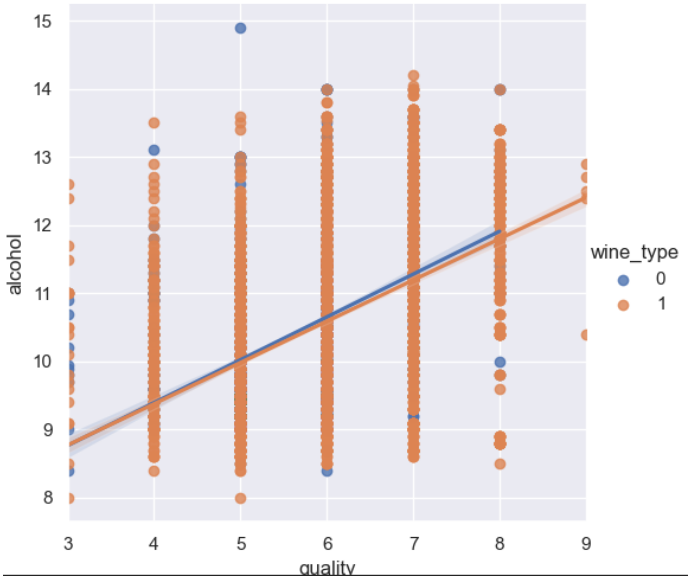
Pandas također pruža funkciju koja vraća statističke podatke o skupu uključujući broj uzoraka, srednju vrijednost, standardnu devijaciju, najmanju vrijednost, vrijednosti prvog, drugog i trećeg kvartila, te najveću vrijednost.

Tablica 4.4. Statistički opis baze podataka

	Fiksna kiselost	Hlapljiva kiselost	Limunske kiseline	Neprevreli šećer	Kloridi	Slobodni sumpor-dioksid	Ukupni sumpor-dioksid	Gustoća	pH	Sulfati	Alkohol	Kvaliteta	Vrsta vina
Zbroj	6497	6497	6497	6497	6497	6497	6497	6497	6497	6497	6497	6497	6497
Srednja vrijednost	7.21	0.33	0.31	5.44	0.05	30.52	115.0	0.99	3.21	0.53	10.49	5.81	0.75
Stand. Devijacija	1.29	0.16	0.15	4.76	0.03	17.45	56.0	0.00	0.16	0.14	1.19	0.87	0.43
1. kvartil	6.40	0.23	0.25	1.80	0.03	177	77.0	0.99	3.11	0.43	8.00	5.00	0.00
2. kvartil	7.00	0.29	0.31	3.00	0.05	29.0	118.0	0.99	3.21	0.51	9.50	6.00	1.00
3. kvartil	7.70	0.40	0.39	8.10	0.07	41.00	156.00	0.99	3.32	0.60	11.30	6.00	1.00
Najveća vrijednost	15.90	1.58	1.66	65.80	0.61	289.00	440.00	1.04	4.01	2.00	14.90	9.00	1.00

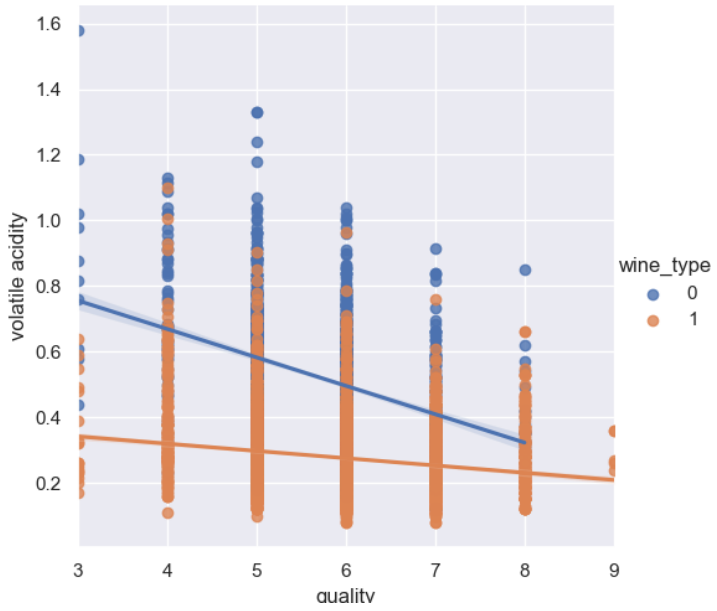
Na neka svojstva u vinu moguće je utjecati. Ako novo vino ima svojstva koja rezultiraju lošom kvalitetom, tada je moguće mijenjati svaku ulaznu varijablu u kasnije napravljenom modelu te testirati kakva bi izlazna kvaliteta bila. Nakon pogođene uspješne kombinacije svojstava, ta svojstva moguće je promijeniti na stvarnom vinu te popraviti njegovu kvalitetu. Na primjer, ako model pokazuje da bi veća razina alkohola povećala kvalitetu, tada bi se proizvođač mogao pobrinuti da temperatura fermentacije bude veća kako bi se podigla razina alkohola. Na isti način moglo bi se utjecati i na druga svojstva.

Iz vizualizacije ovisnosti kvalitete o određenim svojstvima pomoću biblioteke matplotlib moguće je zaključiti kako pojedina svojstva utječu na kvalitetu.



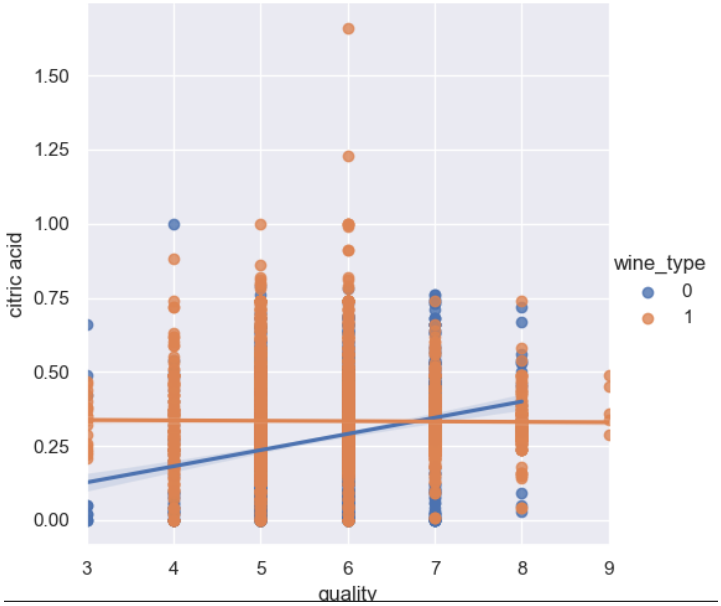
Slika 4.1. Ovisnost kvalitete o količini alkohola

Kao što slika 4.1. pokazuje, vina s većom količinom alkohola u prosjeku znače veću kvalitetu vina.



Slika 4.2. Ovisnost kvalitete o hlapljivoj kiselosti.

Slika 4.2. pokazuje da ako bi vino imalo visoku razinu hlapljive kiselosti da bi to smanjivalo njegovu kvalitetu. U tom slučaju, proizvođač može pokušati smanjiti kiselost da bi povećao kvalitetu.



Slika 4.3. Ovisnost kvalitete o limunskoj kiselini.

Međutim, kod crnih vina (vina označena s 0), rast razine limunske kiseline povećava kvalitetu te bi proizvođač nju naknadno mogao dodati da poboljša vino i to se vidi na slici 4.3. Kod bijelih vina, limunska kiselina ne utječe značajno na kvalitetu.

5. Eksperimentalna evaluacija

U nastavku je opisana implementacija modela opisanih u trećem poglavlju te su prikazani rezultati na način opisan također u trećem poglavlju.

U nastavku će biti prikazana uspješnost oba modela za procjenjivanje vrste i kvalitete vina na testnom skupu. Preciznost oba modela na skupu za učenje iznosi 100% što je očekivano jer to je skup na kojem modeli uče te bilo koja preciznost manja od te ne bi bila zadovoljavajuća.

5.1. Klasifikacija vina ovisno o vrsti

Modeli algoritama slučajne šume i algoritma stroja s potpornim vektorima prvo uče na skupu za učenje da bi ispravno klasificirali vino ovisno o vrsti.

Primjena algoritma slučajne šume pokazana je u prilogu u redcima od 47. do 50**.** Kod slučajne šume potrebno je odrediti koliko će ta šuma sadržavati stabala odluke i to je učinjeno postavljanjem parametra *n_estimators* u retku 47.

Za procjenu vrste pomoću algoritma stroja s potpornim vektorima primijene su Python naredbe u prilogu u redcima 84. do 87.

Tablica 5.1. Matrica zbunjenosti algoritma slučajne šume za vrstu vina.

	Predviđena 0	Predviđena 1
Stvarna 0	335	6
Stvarna 1	1	958

Tablica 5.2. Matrica zbunjenosti algoritma stroja s potpornim vektorima za vrstu vina.

	Predviđena 0	Predviđena 1
Stvarna 0	335	6
Stvarna 1	1	958

Tablica 5.3. Klasifikacijski izvještaj za algoritam slučajne šume za vrstu vina.

	Preciznost	Odziv
Crno vino	1.00	0.98
Bijelo vino	0.99	1.00

Tablica 5.4. Klasifikacijski izvještaj za algoritam stroja s potpornim vektorima za vrstu vina.

	Preciznost	Odziv
Crno vino	1.00	0.98
Bijelo vino	0.99	1.00

Iz tablica 5.1. - 5.4. može se zaključiti kako su oba modela jako uspješna kod prepoznavanja i predviđanja vrste vina. Razlog tome može biti to da su svojstva koja određuju je li vino bijelo ili crno jako izražena.

5.2. Klasifikacija vina ovisno o kvaliteti

Osim klasifikacije ovisno o vrsti, modeli su primijenjeni i za prepoznavanje kvalitete.

Da bi se postigla što veća preciznost moguća, kvalitete vina podijeljene su u tri intervala cijelih brojeva. Vina najlošije kvalitete su ona kvalitete 5 ili manje i odsada su označena s 0.Vina srednje kvalitete su ona s kvalitetom 6 i 7 i označena su s 1. Ostala vina, odnosno vina kvalitete 8 i više su vina visoke kvalitete i označena su s 2.

U prilogu su modeli za učenje kvalitete primijenjeni od 61. do 74. i od 95. do 112. retka. Oba modela koriste OVR pristup koji je opisan u trećem poglavlju.

Tablica 5.5. Matrica zbunjenosti algoritma slučajne šume za kvalitetu vina.

	Predviđena 0	Predviđena 1	Predviđena 2
Stvarna 0	330	121	0
Stvarna 1	111	699	2
Stvarna 2	0	26	11

Tablica 5.6. Matrica zbunjenosti algoritma stroja s potpornim vektorima za kvalitetu vina.

	Predviđena 0	Predviđena 1	Predviđena 2
Stvarna 0	333	118	0
Stvarna 1	111	700	1
Stvarna 2	0	25	12

Tablica 5.7. Klasifikacijski izvještaj za algoritam slučajne šume za kvalitetu vina.

	Preciznost	Odziv
Loša vina	0.75	0.73
Srednja vina	0.83	0.86
Vrhunska vina	0.85	0.30

Tablica 5.8. Klasifikacijski izvještaj za algoritam stroja s potpornim vektorima za kvalitetu vina.

	Preciznost	Odziv
Loša vina	0.75	0.74
Srednja vina	0.83	0.86
Vrhunska vina	0.92	0.32

Iz tablica 5.5. - 5.8. može se zaključiti kako su kod predviđanja kvalitete vina oba klasifikatora postigla lošije rezultate nego kod vrste vina. Algoritam stroja s potpornim vektorima bio je nešto uspješniji, međutim to nije dokaz da je on bolji klasifikator od slučajne šume.

Postoji nekoliko razloga zašto je preciznost kod predviđanja kvalitete manja nego kod predviđanja vrste vina.

Jedan od razloga je to što su svojstva koja određuju kvalitetu manje izražena nego ona kod vrste vina. Kod vrste vina postojala su stroža pravila oko toga kakva svojstva imaju crna, a kakva bijela vina.

Drugi razlog je to što su neke kvalitete manje zastupljene u ovoj bazi podataka pa je modelu teže naučiti koji ulazi rezultiraju tim kvalitetama. Postoji jako malo vina s kvalitetom 9, točnije, samo njih 5 naspram ostalih 6492 vina ostalih kvaliteta. Učenje s takvim malim postotkom zastupljenosti određene kvalitete bilo bi jako

neprecizno ili čak nemoguće. Da bi se to izbjeglo, 9 kvaliteta grupirano je u 3 grupe. Model na taj način više na uči na samo 5 vina kvalitete 9, nego su tu dodana i vina kvalitete 8 kojih ima 193. Međutim, omjer visoke kvalitete naspram ostalih je sada 198 naprema 6299 što je i dalje premala zastupljenost da bi se postigla bolja preciznost.

Zadnji razlog je taj da je kvalitetu teško objektivno procijeniti. Ova baza podataka nastala je testom kušanja. Rezultati takvog testa uvijek su većim dijelom objektivni i ovise o osobi koja vrši test. Kušač može imati specifičan ukus te će dati visoku ocjenu vinu kojemu će druga osoba dati nižu ocjenu. To znači da čak ni stroj ne može uočiti pravilan uzorak kod dobrih vina jer u nekim slučajevima dobra vina odstupaju od tog uzorka. Iako je zbog toga nemoguće postići savršenu preciznost, oba modela dokazala su da je i dalje moguće doseći preciznost prihvatljive razine.

6. Zaključak

Postoji mnogo javno dostupnih baza podataka velikog opsega. Jedna od njih je i baza vina dostupna na internetu koja je korištena u ovom radu. Korištenjem biblioteka u Pythonu, tu bazu bilo je lako oblikovati i pripremiti za strojno učenje.

Algoritam slučajnih šuma i algoritam stroja s potpornim vektorima su se u ovom radu pokazali dobrima za strojno učenje. Njihova preciznost na skupovima za učenje je bila iznimnom visoka, a kod testnih skupova nešto manja, ali i dalje zadovoljavajuća.

To pokazuje da je izbor algoritma slučajnih šuma i algoritam stroja s potpornim vektorima bio dobar te da su oni pogodni za rješavanje ovakvog i sličnog problema.

Također, pronađeno je i više praktičnih primjena ovih klasifikatora.

Izrada ovakvog programa za strojno učenje ne bi imala pravu svrhu kada taj model ne bi mogao biti iskorišten u stvarnom životu. Postoji više načina na koje bi proizvođač ili kupac vina mogao iskoristiti ovakav model.

- Ukoliko proizvođač vina želi da njegovo svako vino ispunjava najviše standarde, može testirati kemijska i fizikalna svojstva svakog proizvedenog bureta. Unošenjem tih svojstava u prethodno naučen model poput ovoga može provjeriti njegovu kvalitetu bez testa kušanja. Ako određeno vino nema dovoljno visoku kvalitetu, tada proizvođač može spriječiti stavljanje vina koje bi razočaralo kupce na police.

- Moguće je napraviti model sličan ovome, ali koristeći drugu bazu podataka. Strasni ljubitelj vina može skupiti sam podatke o kvalitetama vina ovisno o svome ukusu. Naravno da bi bilo teško napraviti bazu ovakvog opsega i veličine, ali bi uz trud bilo moguće napraviti bazu koja bi bila korisna.

- Ovakav model nije ograničen samo na učenje kvalitete i vrsta vina. Vinarija može bilježiti podatke o svakom vinu i uspješnosti njegove prodaje i povratne informacije kupaca. Koristeći te informacije kao izlazne varijable, model bi mogao predvidjeti uspješnost prodaje svake boce prije stavljanja na tržište. Na taj način proizvođač bi mogao više napora uložiti u reklamaciju i marketing boce koja će postići slabije rezultate od prosjeka.

- Kada bi proizvođači u budućnosti pružali više informacije o svakoj boci na etiketi, tada bi svaki kupac mogao unijeti svojstva u model i provjeriti zadovoljava li ta boca ciljanu kvalitetu prije kupnje.

7. Popis literature

[1] P., Cortez a, A., Cerdeira, F., Almeida, T., Matos, J., Reis, *Modeling wine preferences by data mining from physicochemical properties*, Decision Support Systems, br. 47¸ sv. 4, str. 547-533, studeni 2009.

[2] N., Luíza da Costa, L.A., Valentinc , I. A., Castro, R. M., Barbosa, *Predictive modeling for wine authenticity using a machine learning approach*, Artificial Intelligence in Agriculture, br. 5, sv. 1, str. 157-162, srpanj 2021.

[3] Y., Gupta, *Selection of important features and predicting wine quality using machine learning techniques*, 6th International Conference on Smart Computing and Communications, sv. 125, str. 305-312, Kurukshetra, India, 2017.

[4] S., Petropoulosa , C. S., Karavasb , A. T., Balafoutisb , I., Paraskevopoulosd , S., Kallithrakaa, Y., Kotseridis, *Fuzzy logic tool for wine quality classification*, Computers and Electronics in Agriculture, br. 142, sv. B, str. 552-562, studeni 2017.

[5] L., Portinale, G., Leonardi a, M., Arlorio, J. D., Coïsson, F., Travaglia, M., Locatelli, *Authenticity assessment and protection of high-quality Nebbiolo-based Italian wines through machine learning*, Chemometrics and Intelligent Laboratory Systems, br. 171, sv. 1, str. 182-197, prosinac 2017.

[6] S., India, *Best language for Machine Learning: Which Programming Language to Learn* [online], Springboard Blog, 2020., dostupno na: Best language for Machine Learning: Which Programming Language to Learn | Springboard Blog [pristupljeno u srpnju 2021.]

[7] *About pandas* [online], pandas, 2021., dostupno na: pandas - Python Data Analysis Library (pydata.org)

[pristupljeno u srpnju 2021.]

[8] *Introduction* [online], seaborn, 2021., dostupno na: https://seaborn.pydata.org/ [pristupljeno u srpnju 2021.]

[9] *NumPy v1.21 Manual* [online], NumPy, 2021., dostupno na: https://numpy.org/doc/stable/index.html [pristupljeno u srpnju 2021.]

[10] K., Jain, *Scikit-learn(sklearn) in Python – the most important Machine Learning tool I learnt last year!* [online], Analytics Vidhya, 2015., dostupno na: https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/ [pristupljeno u srpnju 2021.]

[11] P., Cortez, A., Cerdeira, F., Almeida, T., Matos and J., Reis, *Wine Quality Data Set* [online], UCL, 2009., dostupno na: https://archive.ics.uci.edu/ml/datasets/wine+quality [pristupljeno u travnju 2021.]

[12] J., Brownlee, *Overfitting and Underfitting With Machine Learning Algorithms* [online], Machine Learning Mastery, 2016., dostupno na: Overfitting and Underfitting With Machine Learning Algorithms (machinelearningmastery.com) [pristupljeno u srpnju 2021.]

[13] J., Brownlee, *Rescaling Data for Machine Learning in Python with Scikit-Learn* [online], Machine Learning Mastery, 2014., dostupno na: https://machinelearningmastery.com/rescaling-data-for-machine-learning-in-python-with-scikit-learn/ [pristupljeno u srpnju 2021.]

[14] S., Ray, *Understanding Support Vector Machine (SVM) algorithm from examples* [online], Analytics Vidhya, 2017., dostupno na: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/ [pristupljeno u srpnju 2021.]

[15] *Understanding the Classification report through sklearn* [online], Muthukrishnan, 2018, dostupno na: Understanding the Classification report through sklearn – Muthukrishnan [pristupljeno u rujnu 2021.]

[16] *OOB Errors for Random Forests* [online], Sci-kit learn, 2020., dostupno na: OOB Errors for Random Forests — scikit-learn 0.24.2 documentation [pristupljeno u srpnju 2021.]

[17] *From a Single Decision Tree to a Random Forest* [online], R. Silipo, dostupno na: From a Single Decision Tree to a Random Forest | by Rosaria Silipo | Towards Data Science [pristupljeno u srpnju 2021.]

Sažetak

Naslov: Procjena kvalitete vina

Kvaliteta vina određena je mnogim kemijsko-fizikalnim svojstvima. Ali i ta svojstva uvjetovana su cijelim procesom nastajanja vina. Kvaliteta plodova vinove loze ovisi o okruženju vinograda i o klimatskim uvjetima kojima je on izložen, poput izloženosti suncu, karakteristike tla i načina berbe. Konkurencija kod proizvodnje vina je velika i potrebno je osigurati konstantno visoku kvalitetu. Ona se provjerava testom kušanja, međutim to nije najefikasnija metoda.

U ovom radu napravit će se nova metoda koja bi bila jeftinija, brža, točnija i efikasnija od testa kušanja. Koristit će se klasifikacija pomoću strojnog učenja (algoritam slučajnih šuma, stroj s potpornim vektorima) kojom će se odrediti kvaliteta vina na temelju jedanaest svojstava. Napravljeni model klasifikatora bi za svako novo proizvedeno vino mogao odrediti njegovu kvalitetu, što znači da bi osiguravao da svako vino koje se stavlja na tržište bude provjereno odlične kvalitete. Također će se zaključiti koja svojstva najviše utječu na kvalitetu te bi se tako svako vino kontroliranjem tih svojstava moglo dodatno poboljšati.

Ključne riječi: kvaliteta vina, algoritam slučajnih šuma, support vector machine, strojno učenje

Abstract

Title: Machine Learning based estimation of wine quality

The quality of wine is determined by many chemical and physical properties, which are conditioned by the entire process of wine production. The quality of the grapevine fruit depends on the environment of the vineyard and the climatic conditions to which it is exposed, such as sun insolation, soil characteristics and harvesting methods. Competition in wine production is high and it is necessary to ensure consistently high quality of produced wine. Wine quality is traditionally checked by tasting, which is not the most effective method.

This paper presents a method that is cheaper, faster, and more efficient than the tasting test. Machine learning (Random Forest Classifier, SVM) is used to determine wine quality based on eleven characteristics. The presented classifier model can determine the quality of each newly produced wine, what ensures high quality of each wine placed on the market. Finally, the most important properties that have impact on quality are emphasized, so that each wine could be further improved by controlling these properties.

Keywords: wine quality, Random forest classifier, Support vectotr machines, machine learning

Životopis

Mihaela Orić rođena je 24. rujna 1999. u Slavonskom Brodu. Osnovnu školu završila je u Slavonskom Kobašu. Nakon osnovne škole, pohađala je Klasičnu gimnaziju fra Marijana Lanosovića s pravom javnosti, Slavonski Brod. Nakon klasične gimnazije upisuje preddiplomski sveučilišni studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija, Osijek.

Prilozi

Programski kod:

```
#Creating the datasets as numpy arrays
X = np.array(wine.drop(labels={'quality','wine_type'}, axis=1))
y_quality = np.array(wine['quality'])
y_type = np.array(wine['wine_type'])
#Classify according to wine type
y = y_type
#Classify according to wine quality
y = y_quality
#Divide data into training/test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, r
andom_state = 42)
#PROVJERA STANDARDIZACIJE
print('*****BEFORE STANDARDIZATION*****')
print(X_train)
print(X_test)
#Standardize dataset (ie mean=0 and std=1)
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
#PROVJERA STANDARDIZACIJE
print('*****AFTER STANDARDIZATION*****')
print(X_train)
print(X_test)
#load red wine dataset
wine_red = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-
databases/wine-quality/winequality-red.csv', sep = ';')
#add column to indicate red wine
wine_red['wine_type'] = 0
#load white wine dataset
wine_white = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-
databases/wine-quality/winequality-white.csv', sep = ';')
#add column to indicate red wine
wine_white['wine_type'] = 1
#Concatenate datasets
wine = pd.concat([wine_red,wine_white], axis=0)
#print first 5 rows
print('*****First 5 rows*****')
print(wine.head())
#print no of instances
```

```
print('*****Wine quality count*****')
print(wine['quality'].value_counts())
print('*****Wine type count*****')
print(wine['wine_type'].value_counts())
#print stats
print('*****Statistics*****')
print(wine.describe())
#Classify according to wine type
y = y_type
print('Random forest classifier')
rfc = RandomForestClassifier(n_estimators = 100)
rfc.fit(X_train, y_train)
rfc_train = rfc.predict(X_train)
rfc_pred = rfc.predict(X_test)
print('Train set')
print(classification_report(y_train, rfc_train))
print(confusion_matrix(y_train, rfc_train))
print('Test set')
print(classification_report(y_test, rfc_pred))
print(confusion_matrix(y_test, rfc_pred))
#Classify according to wine quality
y = y_quality
#Convert wine quality to three levels (low, medium, high)
#low 0-5 -> 0
y[y<=5]=0
#medium 6-7 -> 1
y[(y>=6) & (y<=7)]=1
#high 8-10 -> 2
y[y>=8]=2
#Random forest classifier
print('Random forest classifier')
#initialize rf
rfc = RandomForestClassifier(n_estimators = 100)
#initialize ovr strategy
ovr_rfc = OneVsRestClassifier(rfc).fit(X_train, y_train)
ovr_rfc.fit(X_train, y_train)
ovr_rfc_train = ovr_rfc.predict(X_train)
ovr_rfc_pred = ovr_rfc.predict(X_test)
print('Train set')
print(classification_report(y_train, ovr_rfc_train))
print(confusion_matrix(y_train, ovr_rfc_train))
print('Test set')
print(classification_report(y_test, ovr_rfc_pred))
print(confusion_matrix(y_test, ovr_rfc_pred))
#Classify according to wine type
```

```
y = y_type
print('SVM classifier')
svm = SVC(random_state=32)
svm.fit(X_train, y_train)
svm_train = rfc.predict(X_train)
svm_pred = rfc.predict(X_test)
print('Train set')
print(classification_report(y_train, svm_train))
print(confusion_matrix(y_train, svm_train))
print('Test set')
print(classification_report(y_test, svm_pred))
print(confusion_matrix(y_test, svm_pred))
#Classify according to wine quality
y = y_quality
#Convert wine quality to three levels (low, medium, high)
#low 0-5 -> 0
y[y<=5]=0
#medium 6-7 -> 1
    y[(y>=6) & (y<=7)]=1
    #high 8-10 -> 2
    y[y>=8]=2

    # SVM Classifier
    print('SVM classifier')
    #initialize svm
    svm = SVC(random_state=32)
    #initialize ovr strategy
    ovr_svm = OneVsRestClassifier(rfc).fit(X_train, y_train)
    ovr_svm.fit(X_train, y_train)
    ovr_svm_train = ovr_svm.predict(X_train)
    ovr_svm_pred = ovr_svm.predict(X_test)
    print('Train set')
    print(classification_report(y_train, ovr_svm_train))
    print(confusion_matrix(y_train, ovr_svm_train))
    print('Test set')
    print(classification_report(y_test, ovr_svm_pred))
    print(confusion_matrix(y_test, ovr_svm_pred))
```