Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek

Računarstvo usluga i analiza podataka

Seminarski rad

Klasifikacija lišća

Iva Gavran

Osijek, 2023.

Sadržaj

[Uvod 3](#_Toc142942983)

[Korišteno razvojno okruženje 4](#_Toc142942984)

[Opis programskog rješenja 5](#_Toc142942985)

[Zaključak 13](#_Toc142942986)

# Uvod

Klasifikacija slika je proces pridruživanja određenih kategorija slikama putem strojnog učenja. Cilj klasifikacije je pomoću računalnih algoritama naučiti određeni model da prepozna kojoj kategoriji pripada određena slika. Ta tehnika ima brojne primjene, od identificiranja specifičnih objekata, identifikacije bolesti na medicinskim slikama, detekcije određenih aktivnosti ili ponašanja, do prepoznavanja prometnih znakova i slično. Klasifikacija uključuje prikupljanje te pripremu skupa podataka, izgradnju i treniranje modela te evaluaciju samog modela.

U ovom seminaru obradit ćemo tehniku klasifikacije lišća na temelju skupa podataka preuzetog sa stranice Kaggle.com. Podatci su podijeljeni u dvije skupine:

1. Train skupina – svaka slika je imenovana te uz to ima definirane mjere margina, obrisa te tekstura
2. Test skupina – slike imaju definirane mjere margina, obrisa i tekstura, no nisu imenovane

Train skupina koristi se za treniranje modela, dok se test skupina koristi za evaluaciju. 990 slika nalazi se u skupini za trening, dok se 594 slike nalaze u testnoj skupini, to jest 62.5 % slika predodređeno je za trening, a 37.5 % za testiranje. Budući da se većinom oko 70 % slika predodredi za trening skupinu, možemo reći da je to standardna podjela. Nakon što je model treniran, pomoću test skupine se procjenjuje njegova sposobnost klasifikacije.

A screenshot of a computer

Description automatically generated

Slika 1.1 Korišteni podatci[[1]](#footnote-1)

# Korišteno razvojno okruženje

Algoritam je pisan u JupyterLabu. To je interaktivno razvojno okruženje koje između ostalog omogućuje analizu podataka, izvođenje ekperimenata, kodiranje te stvaranje interaktivnih bilježnica, takozvanih Notebook-a. JupiterLab je nastao iz izvornog Jupytera korištenog uglavnom za kreiranje bilježnica koje su omogućavale kombinaciju kodiranja, teksta, matematičkih jednadžbi te vizualizacije.

Pomoću JupyterLab-a, omogućen je primjerice rad u više prozora, pregled datotečnog sadržaja te pokretanje koda u sekvencama u bilježnicama čiji izlaz može biti, osim u obliku teksta, u obliku grafova, vizualizacija i slično. JupyterLab omogućava integraciju različitih vizualizacijskih alata kako bi se podatci prezentirali što učinkovitije.

Ovo razvojno okruženje često se koristi u znanstvenim istraživanjima, analizi podataka te strojnom učenju.

Kod je pisan u Python programskom jeziku, no JupyterLab podržava i ostale programske jezike poput R-a te Julie. Na slici ispod je prikaz JupyterLaba. [[2]](#footnote-2)

A screenshot of a computer

Description automatically generated

Slika 2.1 JupyterLab

# Opis programskog rješenja

Programsko rješenje započelo je uvođenjem biblioteke pandas. Pandas je popularna open-source Python biblioteka koja se koristi za analizu podataka. Ključna je komponenta u istraživanju podataka, njihovoj pripremi za strojno učenje te u radu s tabličnim strukturama. Omogućava razne operacije nad podatcima poput filtriranja, agregacije, transformacije te vizualizacije.

U našem slučaju, korištena je za čitanje csv datoteke u kojoj se nalaze podatci za treniranje. Pomoću train.head(10), ispisali smo prvih 10 podataka kako bi provjerili ispravnost učitane datoteke:

A screenshot of a computer

Description automatically generated

Nakon toga, bilo je potrebno uključiti određene module sklearn biblioteke. To je također popularna open-source biblioteka za strojno učenje koja pruža velik broj alata, funkcionalnosti i algoritama za razvoj i primjenu strojnog učenja. Prvo je uključena klasa LabelEncoder unutar sklearn.preprocessing modula kako bi se tekstualne vrijednosti stupca species pretvorile u numeričke. Te numeričke vrijednosti spremljene su u varijablu species, a podatci, s uklonjenim stupcima id i species, u varijablu data:

A screenshot of a computer code

Description automatically generated

Zatim se pomoću StandardScaler klase za svaki podatak izračunala srednja vrijednost i standardna devijacija te se metodom transform to primijenilo na podatke. Takvo skaliranje centrira podatke oko nule te ih skalira prema standardnoj devijaciji. Nakon toga, pomoću MinMaxScaler klase se osiguralo da nema negativnih vrijednosti te su se nakon te obrade podatci podijelili na skupove za treniranje i testiranje budući da ćemo prvo istestirati ovaj model na podatcima za treniranje čije su slike imenovane kako bi provjerili kvalitetu algoritma. 30 % podataka za treniranje odabrano je za testni uzorak.

A screenshot of a computer

Description automatically generated

Zatim se definirao SVM (Support Vector Machine) model gdje parametar C iznosi 50, gamma 0.01, a kernel, to jest jezgra je rbf. . Zatim se model trenirao.

Nakon toga, pomoću decision\_function() metode vratila se vrijednost za svaki uzorak. Metodom predict() odrađeno je predviđanje temeljeno na naučenom modelu. Kako se može vidjeti, točnost je 98.99 %:  
  
A screenshot of a computer

Description automatically generated

Sada smo mogli primijeniti taj model na prave testne podatke iz testne datoteke koji nisu bili imenovani te koje je naš model tek trebao prepoznati te zatim imenovati. Počeli smo s učitavanjem podataka za treniranje te ponovili korake, no ovog puta nismo dijelili podatke za trening. SVC-u je dodan i argument probability kako bi se uz predviđanja, uključila i vjerojatnost da je dani rezultat točan:

A screen shot of a computer code

Description automatically generated

Zatim smo učitali testnu csv datoteku koja nema imenovane vrste lišća, to jest stupac species, već samo id-eve koji govore o kojoj slici iz datoteke images se radi. Nakon toga smo obradili podatke kao što je to učinjeno s podatcima za trening. Klase i id-eve smo spremili u varijable kako bi ih mogli kasnije koristiti:

A screenshot of a graph

Description automatically generated

A white background with black text

Description automatically generated

Nakon što su podatci obrađeni, pomoću metode predict\_proba() dobili smo matricu vjerojatnosti, to jest vjerojatnost da određena slika pripada određenoj vrsti (species).

A screenshot of a computer

Description automatically generated

Nakon sređivanja podataka, dobili smo sljedeću tablicu:

A screenshot of a computer

Description automatically generated

Sada smo za svaku testnu sliku imali informaciju kolika je vjerojatnost da pripada određenoj vrsti lišća. Ispisali smo sve vjerojatnosti u csv file, a zatim te podatke uredili tako što smo za svaku vrstu ispisali sve id-eve koji za tu vrstu imaju maksimalnu vjerojatnost, to jest najvjerojatnije joj pripadaju, i tako odradili klasifikaciju. Te podatke smo spremili u csv file classification.csv. Kako bi popis bio pregledniji, imena vrsta smo abecedno sortirali:

A screenshot of a computer program

Description automatically generated

Na samom kraju, kako bi se uvjerili da je klasifikacija dobro odrađena, testne slike su prikazane vizualno za svaku vrstu. Nakon njih, slike iz trening skupine su također prikazane vizualno kako bi se mogla usporediti točnost klasifikacije:

A computer screen shot of a computer

Description automatically generated

A computer code on a white background

Description automatically generated

Na lijevoj strani prikazano je prvih deset vrsta lišća iz testne skupine, a na desnoj strani prvih deset vrsta iz trening skupine.

A screenshot of a computer

Description automatically generatedA chart of leaves and a foot

Description automatically generated with medium confidence

# Zaključak

Kao što se može vidjeti na temelju slika iz testne skupine, klasifikacija je uspješno odrađena, no ne i savršeno. Poneke vrste lišća nisu apsolutno prepoznate, primjerice Acer Opalus te Acer Rubrum, no većinu vrsta je algoritam uspješno prepoznao te samim time vrlo dobro odradio klasifikaciju. Budući da je većina slika iz testne skupine dobro klasificirana, možemo zaključiti da je algoritam zadovoljio očekivanja, ali i da ima prostora za njegov napredak.

1. Podatci, kao i slika, preuzeti su sa stranice Kaggle.com [↑](#footnote-ref-1)
2. Slika je preuzeta sa stranice https://blog.jupyter.org/jupyterlab-is-ready-for-users-5a6f039b8906 [↑](#footnote-ref-2)