**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA**

Klasifikacija filmskog žanra u odnosu na opis

Kolegij: Raspoznavanje uzoraka i strojno učenje

**Leon Šumanovac**

**Osijek, 2023.**

**SADRŽAJ**

[**1.** **UVOD** 3](#_Toc138607792)

[**2.** **ISTRAŽIVAČKA ANALIZA PODATAKA** 4](#_Toc138607793)

[**2.1.** **Korištene biblioteke** 5](#_Toc138607794)

[**2.2.** **Priprema podataka** 6](#_Toc138607795)

[**2.3.** **Predikcija podataka** 6](#_Toc138607796)

[**3.** **KORIŠTENI ALGORITMI – TEORIJSKI OPIS** 7](#_Toc138607797)

[**2.1.** **Logistička regresija** 7](#_Toc138607798)

[**2.2.** **LinearSVC** 7](#_Toc138607799)

[**3.1.1.** **Matrica zabune** 8](#_Toc138607800)

[**3.** **APLIKACIJA ZA PRIKAZ REZULTATA – STREAMLIT** 9](#_Toc138607801)

[**4.** **ZAKLJUČAK** 14](#_Toc138607802)

# **UVOD**

Ovaj projekt će se baviti problemom klasifikacije filmskog žanra iz dobivenog opisa filma. Klasifikacija će biti odrađena pomoću dva algoritma, a to su linearna regresija i LinearSVC (Linear Support Vector Classifier).

Projekt nudi mogućnost unošenja vlastitog opisa filma, te aplikacija pomoću Streamlit-a koristi zadnji korišteni algoritam, te kroz njega „provlači“ uneseni opis i ispisuje rezultat.

Za ovaj projekt će se koristit Python 3.9.

# **ISTRAŽIVAČKA ANALIZA PODATAKA**

Istraživačka analiza podataka odnosi se na kritičan proces izvođenja početnih istraživanja podataka kako bi se otkrili obrasci, uočile anomalije, testirale hipoteze i provjerile pretpostavke uz pomoć zbirne statistike i grafičkih prikaza.

Dataset koji je korišten može se pronaći na ovome linku: <https://www.kaggle.com/datasets/hijest/genre-classification-dataset-imdb>

Unutar njega zabilježeni su sljedeći podaci o filmovima: id, naslov (s godinom izdanja), žanr i opis.

Podaci su spremljeni u .txt file sa separatorom ':::'.

A screen shot of a computer

Description automatically generated with medium confidence

**Slika 2.1** Prikaz parametara dataseta

## **Korištene biblioteke**

Unutar rada korištene su biblioteke:

1. Pandas
2. Numpy
3. Streamlit
4. Matplotlib
5. Scikit-learn
6. Pickle
7. Os

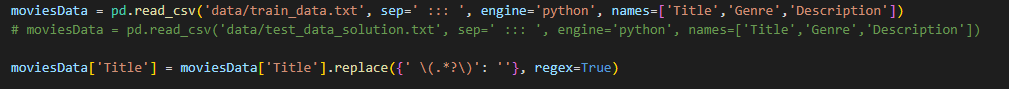
A screen shot of a computer program

Description automatically generated with low confidence

**Slika 2.1.1** Prikaz korištenih biblioteka

Nakon što su biblioteke uvezene u program, podatci se učitavaju iz dataseta train\_data.txt koji se nalazi u folderu data. U varijablu se spremaju stupci pod nazivima Title, Genre i Description razdvojeni sa ':::' kao što je to rečeno u prethodnom poglavlju. Podatci su spremljeni u varijablu tipa DataFrame.

Nakon uspješnog učitavanja podataka, brišemo godinu izdavanja iz naslova filma.



**Slika 2.1.2** Prikaz učitavanja podataka i brisanja godine iz naslova

## **Priprema podataka**

Kako bi koristili podatke iz dataseta potrebno ih je obraditi i pripremiti kako bi ih model mogao koristiti zajedno.

Kako bi to postignuli kreirana je funckija preprocessData koja vraća features i target. I features i target su tipa list.

A screen shot of a computer code

Description automatically generated with low confidence

**Slika 2.2.1** Prikaz pripreme podataka za korištenje u modelu

## **Predikcija podataka**

Nakon što su podatci pripremljeni, definirana je funkcija predict, u kojojj se poziva priprema podataka i obrada samih podataka.

A picture containing text, screenshot, software

Description automatically generatedFunkcija na početku pomoću biblioteke os briše stari dump ako on postoji od prethodnog poziva funkcije predict. Nakon toga poziva pripremu podataka i dijeli ih na trening podatke i na test podatke u omjeru 0.8 – 0.2.

**Slika 2.3.1** Funkcija za predikciju

Nakon toga se bira model algoritma po kojim će se nastaviti vršiti funkcija. Model se bira preko primljenog parametra iz main-a. Nakon toga su podatci fitani koristeći model zadanog algoritma, te je odrađena predikcija nad testnim podatcima. Dump podaci o modelu su spremljeni pomoću biblioteke Pickle, te se izvršava prikaz matrice zabune i točnosti na Streamlit stranici.

# **KORIŠTENI ALGORITMI – TEORIJSKI OPIS**

Nakon prikazanih podataka ćemo ponešto reći o svakome algoritmu od kojega ćemo dobiti rezultate i prikazati točnost pojedinog algoritma

Algoritmi koji su korišteni:

1. LinearSVC
2. LogisticRegression

Koristili smo algoritme u svrhu izračuna podataka te smo dobivene vrijednosti stavili u matricu zabune kako bi dobili određenu točnost za pojedini algoritam.

## **Logistička regresija**

Logistička regresija je model klasifikacije, a ne regresijski model unatoč svog naziva. Logistička regresija je jednostavna i učinkovitija metoda za probleme binarne i linearne klasifikacije. To je klasifikacijski model koji je vrlo lako realizirati i postiže vrlo dobre performanse s linearno odvojivim klasama. Ona je vrlo opsežno korišten algoritam za klasifikaciju u industriji. Logistički regresijski model, poput Adaline i perceptrona, statistička je metoda za binarnu klasifikaciju koja se može generalizirati na višeklasnu klasifikaciju. Scikit-learn ima vrlo optimiziranu verziju implementacije logističke regresije, koja podržava zadatak klasifikacije više klasa.

## **LinearSVC**

LinearSVC je linearni algoritam modela za klasifikaciju temeljen na SVM (Support Vector Machine). Njegov cilj je pronalazak optimalne hiperravnine kako bi ona razdvojila podatke različitih klasa. Često se koristi „One vs all“ pristup, u kojemu se svaka klasa razdvaja od ostalih. Ovaj je algoritam vrlo efikasan pri obradi podataka visoke dimenzionalnosti i velikih skupova podataka.

### **Matrica zabune**

A screenshot of a game

Description automatically generated with medium confidenceMatrica konfuzije sadrži informacije o očekivanim i pretpostavljenim vrijednostima našeg modela gdje redci predstavljaju pretpostavljenu vrijednost a stupci očekivanu. Glavna dijagonala predstavlja točno predpostavljene elemente.

**Slika 3.1.1.1** Matrica konfuzije

Također, pomoću matrice konfuzije možemo također izračunati i točnost, osjetljivost i preciznost. Točnost je omjer točno klasificiranih primjera u odnosu na ukupan broj primjera. Pozitivni primjeri su nam daleko važniji (medicina, točnost daje istu težinu i pozitivnim i negativnim primjerima) te također u mnogim problemima imamo veliki nesrazmjer između broja pozitivnih i negativnih primjera.

A screenshot of a computer

Description automatically generated with medium confidenceU ovom projektu, na algoritmu logističke regresije dobijamo ovakav primjer:

**Slika 3.1.1.2** Matrica dobivena algoritmom logističke regresije

Uslikan je samo dio matrice zbog preglednosti (matrica zabune iz projekta ima 27 stupaca i redaka).

# **APLIKACIJA ZA PRIKAZ REZULTATA – STREAMLIT**

Za prikaz rezultata modela se koristila Python-ova biblioteka Streamlit koja omogućuje podizanje web aplikacije.

Aplikaciju pokrećemo naredbom: streamlit run main.py

Nakon toga se podiže web sučelje, pali se dafault-ni browser na localhost:8501 i u terminalu VS Code-a se dobiva sljedeća povratna informacija: A screen shot of a computer

Description automatically generated with medium confidence

**Slika 4.1** Podizanje web alikacije

Da bi na stranici išta bilo prikazano, moramo to napisati u python skripti. Slika 4.2 prikazuje implementaciju main funkcije ovog projekta.

A screen shot of a computer screen

Description automatically generated with low confidence

**Slika 4.2** Prikaz implementacije izgleda web sučelja

Web sučelje koje se pojavljuje izgleda kao prikazano na slici, uz otvorenu početnu stranicu Singular predistion.

A screenshot of a movie category

Description automatically generated with medium confidence

**Slika 4.3** Izgled web sučelja

Prilikom unošenja opisa filma i klikom na submit, dobiva se predikcija žanra filma. Za sljedeću fotografiju je unesen opis filma Flash (2023), te je očekivani rezultat Action / Adventure / Fantasy. Očekivani rezultat i opis se nalaze na sljedećem URL-u:  
 <https://www.imdb.com/title/tt0439572/>

A screenshot of a computer

Description automatically generated with medium confidence

**Slika 4.4** Prikaz rezultata za film Flash (2023)

Klikom na gumb „Movie categories“ se prikazuje sljedeći graf:

A screen shot of a graph

Description automatically generated with medium confidence

**Slika 4.5** Prikaz podataka za filmske žanrove u datasetu

Klikom na gumb LinearSVC se prikazuje sljedeća matrica zabune s pripadnom točnosti:

A picture containing text, screenshot, font, number

Description automatically generated

**Slika 4.6** LinearSVC s pripadajućom točnosti

Iz slike 4.6 se vidi da LinearSVC ima točnost od oko 51.5%. To je iz razloga loše raspodjeljenosti klasa u datasetu. Također, većina filmova se može svrstat u više od jednog žanra, a sukladno s time i više od jedne klase, no dataset korišten za ovaj projekt ima samo jedan žanr po filmu.

Klikom na zadnji gumb „Logistic Regression“ se prikazuje sljedeći graf:

A screen shot of a computer

Description automatically generated with low confidence

**Slika 4.7** Logistic regression s pripadajućom točnosti

Iz slike 4.7 se vidi da je točnost Logističke regresije oko 57.8%. Točnost je veća u odnosu na točnost od LinearSVC, no i dalje je mala, a to je i dalje problem dataset-a koji je objašnjen kod slike 4.6.

# **ZAKLJUČAK**

Korištenje prepoznavanja uzoraka i strojnog učenja se može koristiti za prikazivanje žanra filma ovisno o opisu istog, samo je potrebnoi imati podjednako raspoređenu količinu klasa i popis svih žanrova filma u tom datasetu.

Algoritam logističke regresije se ukazao boljim za oko 6.5% od LinearSVC algoritma.