

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

**Sveučilišni diplomski studij**

**APLIKACIJA ZA DETEKCIJU JEZIKA**

**Raspoznavanje uzoraka i strojno učenje**

**Mario Pandurić**

**Osijek, 2023.**

# SADRŽAJ

1.	UVOD.....	1
2.	KORIŠTENE TEHNOLOGIJE.....	2
2.1.	Anaconda .....	2
2.2.	Streamlit.....	2
2.3.	Popis korištenih biblioteka.....	3
3.	KORIŠTENI ALGORITMI STROJNOG UČENJA.....	4
3.1.	Naivni Bayesov klasifikator .....	4
3.2.	Logistička regresija.....	5
3.3.	K najbližih susjeda.....	5
3.4.	Strojevi s potpornim vektorima.....	5
3.5.	Slučajne šume.....	6
4.	KORIŠTENI SKUP PODATAKA.....	7
5.	STREAMLIT APLIKACIJA .....	9
6.	TESTIRANJE I PRIKAZ METRIKA .....	11
6.1.	Testiranje i prikaz rezultata Naivnog Bayes algoritma.....	12
6.2.	Testiranje i prikaz rezultata logističke regresije.....	15
6.3.	Testiranje i prikaz rezultata K najbližih susjeda.....	18
6.4.	Testiranje i prikaz rezultata strojeva s potpornim vektorima.....	21
6.5.	Testiranje i prikaz rezultata algoritma slučajne šume.....	24
7.	ZAKLJUČAK.....	28
8.	LITERATURA.....	29

# 1. UVOD

Jezik je najosnovniji način komuniciranja kod ljudi. Sastoji se od niza simbola s određenim značenjem i niza pravila kojim te simbole povezujemo. U svijetu postoji oko 6800 različitih jezika.

U ovom projektu će se pomoću različitih algoritama strojnog učenja klasificiranja, na nekom tekstu, pokušati odrediti na kojem je jeziku tekst napisan. Bit će kreirana web aplikacija pomoću Streamlit radnog okvira putem koje će se moći unijeti tekst kojem se želi identificirati jezik. Svakom algoritmu ćemo odrediti metrike evaluacije te ćemo usporediti ih usporediti i odlučiti koji je algoritam najbolji odnosno najlošiji.

## **2. KORIŠTENE TEHNOLOGIJE**

### **2.1. Anaconda**

Anaconda je besplatna i open-source distribucija Pythona i R programskog jezika koja se koristi u znanstvenim istraživanjima, analizi podataka, razvoj softvera i sl. Cilj joj je pojednostaviti upravljanje paketima i implementaciju. Distribucija uključuje pakete korištene za podatkovne znanosti prikladne za Windows, Linux i macOS operacijske sustave. Anaconda dolazi s preko 250 instaliranih paketa te je još moguće instalirati više od preko 7500 paketa. Anaconda dolazi i s grafičkim korisničkim sučeljem „Anaconda Navigator“ kao alternativa za komandno sučelje. Anaconda Navigator omogućuje pokretanje aplikacija i upravljanje paketima bez korištenja komandne linije. Aplikacije koje su dostupne u Anaconda Navigatoru su: JupyterLab, JupyterNotebook, QtConsole, Spyder, Glue, Orange, RStudio, Visual Studio Code. Korištena verzija Anaconde u ovom projektu je 22.9.0.

### **2.2. Streamlit**

Streamlit je open-source biblioteka za Python koja se koristi za brzo i jednostavno razvijanje web aplikacija za vizualizaciju podataka, prikazivanje modela strojnog učenja i analize podataka. Podržava mnoge popularne biblioteke za vizualizaciju podataka i strojno učenje kao što su npr. Pandas, Matplotlib, Plotly, scikit-learn i TensorFlow što programerima omogućuje da jednostavno uključe svoje postojeće analize podataka ili modele strojnog učenja u interaktivne aplikacije. Streamlit omogućuje dijeljenje aplikacija putem javnih URL-ova. Verzija korištena u ovom projektu je 1.23.1.

## 2.3. Popis korištenih biblioteka

Korištene Python biblioteke za izradu projekta su:

- Streamlit
- Pandas
- Numpy
- Matplotlib
- Scikit-Learn
- Time

```
import streamlit as st
import pandas as pd
import numpy as np
from sklearn.metrics import classification_report, precision_score, recall_score, f1_score, confusion_matrix, accuracy_score, ConfusionMatrixDisplay
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn import svm
from sklearn.svm import LinearSVC
from sklearn.ensemble import RandomForestClassifier
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
import time
```

*Slika 2.3.1. Prikaz korištenih biblioteka u kodu*

### 3. KORIŠTENI ALGORITMI STROJNOG UČENJA

#### 3.1. Naivni Bayesov klasifikator

Naivni Bayesov klasifikator je vjerojatnosni klasifikator koji se temelji na Bayesovom teoremu. Bayesov teorem opisuje vjerojatnost da se dogodi neki događaj A ako znamo da se dogodio neki događaj B.

Jednadžba Bayesovog teorema:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}, \text{ gdje je:}$$

- $P(A|B)$  – vjerojatnost događaja A ako se dogodio događaj B
- $P(B|A)$  – vjerojatnost događaja B ako se dogodio događaj A
- $P(A)$  – vjerojatnost događaja A
- $P(B)$  – vjerojatnost događaja B

Koristeći ovaj teorem, moguće je odrediti klasifikaciju neke instance na temelju vjerojatnosti njezinih atributa. Naivni Bayesov algoritam je osjetljiv na pretpostavku o nezavisnosti atributa odnosno pretpostavlja da je pojavljivanje svakog atributa neovisno o ostalima.

### **3.2. Logistička regresija**

Logistička regresija predstavlja statistički model koji se koristi za predviđanje binarne ili višeklasne ovisne varijable na temelju linearnih kombinacija nezavisnih varijabli. Linearnu logističku regresiju karakterizira linearna kombinacija ulaznih atributa. Svaki primjer se predstavlja kao vektor atributa, a linearna kombinacija je suma težinskih koeficijenata pomnoženih s odgovarajućim atributima. Ova suma se zatim prolazi kroz logističku funkciju.

### **3.3. K najbližih susjeda**

K najbližih susjeda (KNN) je algoritam klasifikacije ili regresije koji se temelji na ideji da se slični uzorci grupiraju zajedno. Na početku je potrebno odrediti parametar K koji predstavlja broj najbližih susjeda koji će algoritam uzeti u obzir kod donošenja odluke. Za svaki novi uzorak se računa udaljenost (Euklidska ili neka druga) njegovih susjeda te se određuju najbliži susjedi. Algoritam donosi predikciju na temelju njihovih srednjih vrijednosti (regresija) ili na temelju njihovih klasa (klasifikacija).

### **3.4. Strojevi s potpornim vektorima**

Strojevi s potpornim vektorima (SVM) je algoritam strojnog učenja koji se koristi za binarnu ili višeklasnu klasifikaciju i regresiju. Cilj SVM-a je pronaći hiperravninu (u 2D prostoru liniju, u 3D prostoru ravninu) koja odvaja uzorke različitih klasa. Ona se pronalazi na način da je maksimalno udaljena od najbližih uzoraka svake klase.

### **3.5. Slučajne šume**

Algoritam slučajnih šuma je algoritam koji predikciju nekog uzorka temelji na stablu odlučivanja. Nakon što su algoritmu predani podaci s kojima će raditi, postupak odlučivanja se svodi na niz postavljanja pravih pitanja. Odgovori na pitanja su u obliku točno-netočno ili 0-1. Svaki krajnji čvor stabla predstavlja određenu klasu.



## 4. KORIŠTENI SKUP PODATAKA

Korišteni skup podataka preuzet je sa stranice Kaggle [1]. Skup podataka sadrži 22 jezika. Sastoji se od nekoliko rečenica koje su zapisane u jedan redak. U istom retku, ali idućem stupcu je zapisan jezik kojem te rečenice pripadaju. Ukupno je 22000 redova odnosno po 1000 za svaki jezik.

U projektu je za klasifikaciju korišteno 11 jezika: engleski, estonski, nizozemski, latinski, španjolski, portugalski, švedski, rumunjski, indonezijski, francuski i turski.

1	Text	language
2	klement gottwalddi surnukeha palsam	Estonian
3	sebes joseph pereira thomas på eng	Swedish
4	ถนนเจริญกรุง อักษรโรมัน thanon charo	Thai
5	விசாகப் பட்டினம் தமிழ்ச்சங்க	Tamil
6	de spons behoort tot het geslacht hal	Dutch
7	エノが行きがかりでバスに乗っ	Japanese
8	tsutınalar ingilizce tsuutina kanadada	Turkish
9	müller mox figura centralis circulatorur	Latin
10	تمام زیرجوہری ذرات electric charge برقی بار	Urdu
11	シャーリー・フィールドは、サ	Japanese
12	kemunculan pertamanya adalah ketika	Indonesian
13	barocco pt escândalo de a página é ur	Portugese
14	association de recherche et de sauve	French
15	胡赛尼本人和小说的主人公阿米尔	Chinese
16	한국에서 성씨가 사용되기 시작	Korean
17	การฟาดฟันของบรรดาสาวงามในกองป	Thai
18	dorota rabczewska artistinimega dod	Estonian
19	diante destes gerenciamentos poden	Portugese

Slika 4.1. Prikaz dijela skupa podataka

Nakon importa podataka u Python, podaci su prebačeni u „DataFrame“ strukturu. Nakon toga su uklonjeni određeni jezici te su dodane labelle i podaci su podijeljeni u omjeru 80 : 20 za trening i testiranje.

```

data = pd.read_csv(r"dataset.csv")
data = pd.DataFrame(data)

data = data[data["language"].str.contains("Chinese") == False]
data = data[data["language"].str.contains("Thai") == False]
data = data[data["language"].str.contains("Persian") == False]
data = data[data["language"].str.contains("Japanese") == False]
data = data[data["language"].str.contains("Hindi") == False]
data = data[data["language"].str.contains("Pushto") == False]
data = data[data["language"].str.contains("Arabic") == False]
data = data[data["language"].str.contains("Tamil") == False]
data = data[data["language"].str.contains("Urdu") == False]
data = data[data["language"].str.contains("Russian") == False]
data = data[data["language"].str.contains("Korean") == False]

x = np.array(data["Text"])
y = np.array(data["language"])

cv = CountVectorizer()
X = cv.fit_transform(x)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)

```

Slika 4.2. Prikaz dijela koda kod importa i obrade podataka

## 5. STREAMLIT APLIKACIJA

Aplikacija se sastoji od od 5 gumbova za pojedini algoritam i jedan gumb kojim se uspoređuju svih 5 algoritama, njihove metrike i brzina izvođenja. Također se nalazi jedan „text input“ za unos željenog teksta kojem se želi identificirati jezik. Nakon unosa teksta klikom na jedan od gumbova algoritama prikazuje se matrica zabune te metrike za evaluaciju. Klikom na gumb „compare“ pojavljuju se linijsku i stupčasti dijagram. Linijski dijagram predstavlja trajanje izvođenja svakog algoritma dok stupčasti dijagram predstavlja metrike svakog algoritma.

```
def showResult(model):
    model.fit(X_train, y_train)
    output = model.predict(cv.transform([text]).toarray())
    st.write("Result: " + str(output))
    labels = ['Dutch', 'English', 'Estonian', 'French', 'Indonesian', 'Latin', 'Portugese', 'Romanian', 'Spanish', 'Swedish', 'Turkish']
    st.text("Confusion matrix:")
    cmatrix = confusion_matrix(y_test, model.predict(X_test))
    cm_display = ConfusionMatrixDisplay(confusion_matrix=cmatrix, display_labels=labels)
    fig, ax = plt.subplots()
    cm_display.plot(ax=ax, cmap=plt.cm.Blues)
    plt.xticks(rotation=90)
    st.pyplot(fig)
    st.write('Classification report:')
    clreport = classification_report(y_test, model.predict(X_test))
    st.text(clreport)
```

Slika 5.1. Prikaz koda za prikaz matrice zabune i metrike za evaluaciju

```

for model in models:
    start_time = time.time()
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    execution_time = time.time() - start_time
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')

    metrics['Accuracy'].append(accuracy)
    metrics['Precision'].append(precision)
    metrics['Recall'].append(recall)
    metrics['F1 Score'].append(f1)
    metrics['Execution Time'].append(execution_time)

fig, ax1 = plt.subplots()
ax2 = ax1.twinx()

x = np.arange(len(model_names))
width = 0.2

ax1.bar(x - width, metrics['Accuracy'], width, label='Accuracy')
ax1.bar(x, metrics['Precision'], width, label='Precision')
ax1.bar(x + width, metrics['Recall'], width, label='Recall')
ax1.bar(x + (2 * width), metrics['F1 Score'], width, label='F1 Score')

ax2.plot(x, metrics['Execution Time'], color='blue', marker='o', label='Execution Time')

ax1.set_xlabel('Algorithms')
ax1.set_ylabel('Values')
ax2.set_ylabel('Execution Time (s)')
ax1.set_title('Metrics of algorithms')
ax1.set_xticks(x)
ax1.set_xticklabels(model_names)
ax1.legend(loc='upper left')
ax2.legend(loc='upper right')
plt.pyplot(fig)

```

Slika 5.2. Prikaz dijela koda za uspoređivanje svih algoritama

## 6. TESTIRANJE I PRIKAZ METRIKA

Informacije koje dobijemo nakon predviđanja o svakom algoritmu su: matrica zabune, preciznost i točnost, odziv, f1 rezultat i broj uzoraka u skupu podataka. Matrica zabune prikazuje broj dobro, odnosno broj loše predviđenih klasa.

TP – broj uzoraka koji su ispravno klasificirani kao pozitivni

TN – broj uzoraka koji su ispravno klasificirani kao negativni

FP – broj uzoraka koji su krivo klasificirani kao pozitivni

FN – broj uzoraka koji su negativno klasificirani kao pozitivni

$$preciznost = \frac{TP}{TP + FP}$$

$$odziv = \frac{TP}{TP + FN}$$

$$točnost = \frac{TP + TN}{TP + FP + TN + FN}$$

$$F1 = 2 * \frac{preciznost * odziv}{preciznost + odziv}$$

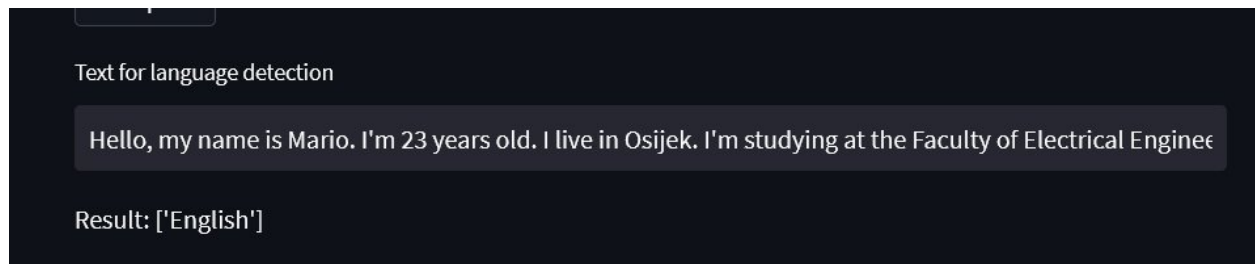
## 6.1. Testiranje i prikaz rezultata Naivnog Bayes algoritma

Za testiranje algoritama uzet je isti tekst na 3 različita jezika: engleski, rumunjski i španjolski.

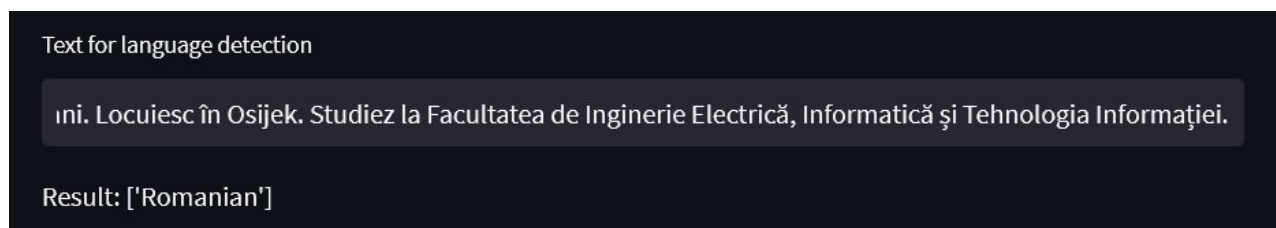
Tekst na engleskom: *Hello, my name is Mario. I'm 23 years old. I live in Osijek. I'm studying at the Faculty of Electrical Engineering, Computer Science, and Information Techonlogy.*

Tekst na rumunjskom: *Bună, numele meu este Mario. Am 23 de ani. Locuiesc în Osijek. Studiez la Facultatea de Inginerie Electrică, Informatică și Tehnologia Informației.*

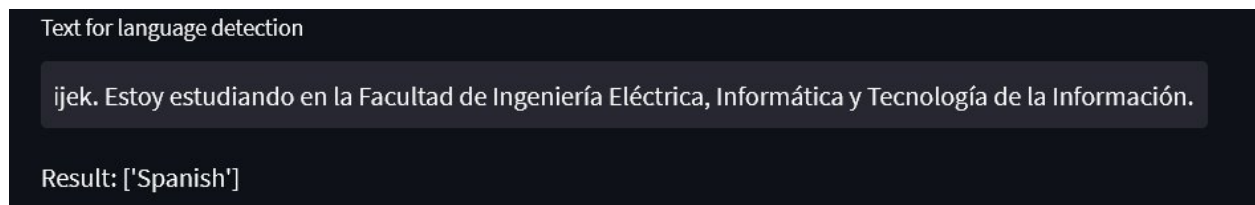
Tekst na španjolskom: *Hola, mi nombre es mario. Tengo 23 años. Vivo en Osijek. Estoy estudiando en la Facultad de Ingeniería Eléctrica, Informática y Tecnología de la Información.*



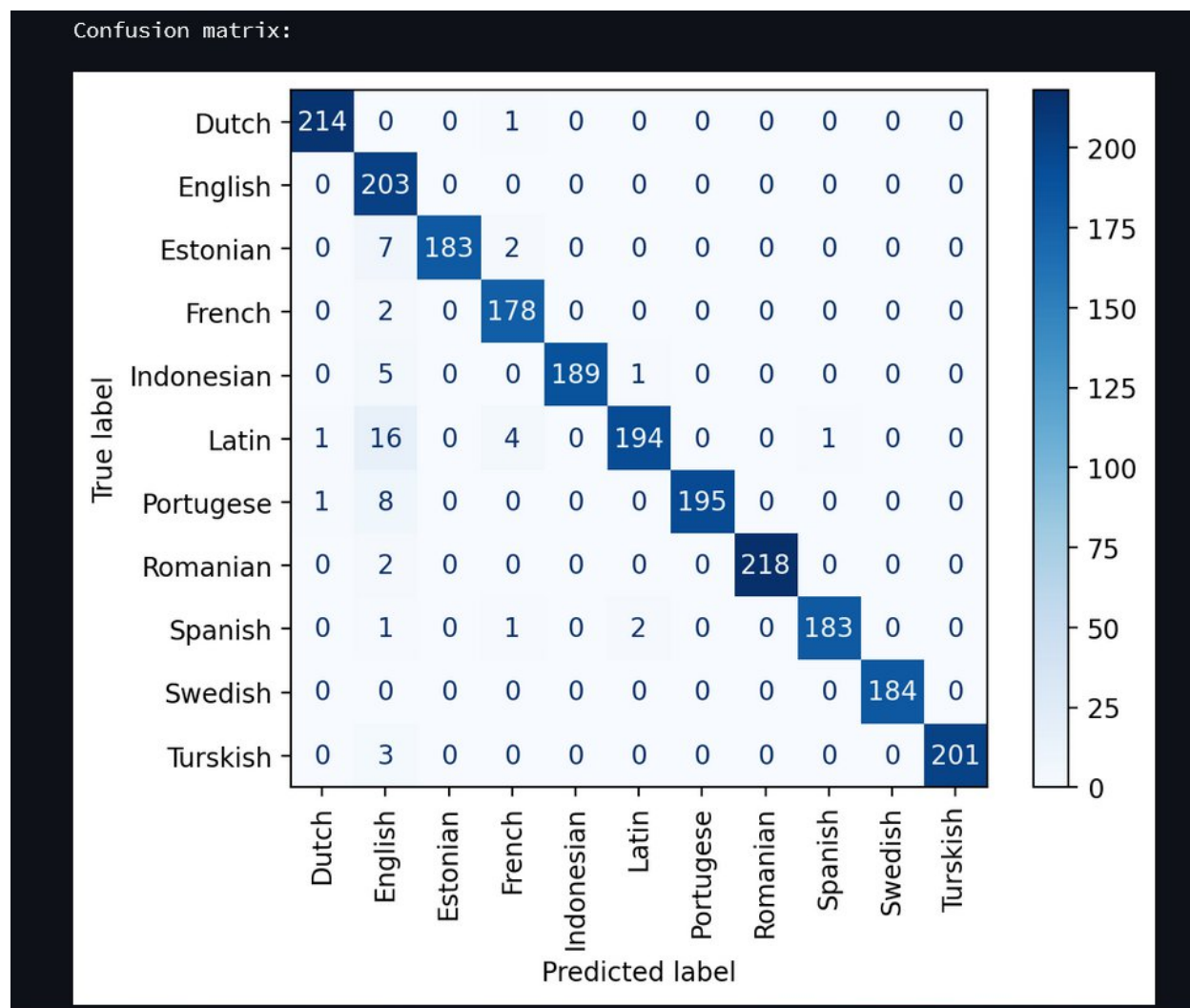
Slika 6.1.1. Prikaz rezultata Naivnog Bayes algoritma za engleski jezik



Slika 6.1.2. Prikaz rezultata Naivnog Bayes algoritma za rumunjski jezik



Slika 6.1.3. Prikaz rezultata Naivnog Bayes algoritma za španjolski jezik



Slika 6.1.4. Prikaz matrice zabune Naivnog Bayes algoritma



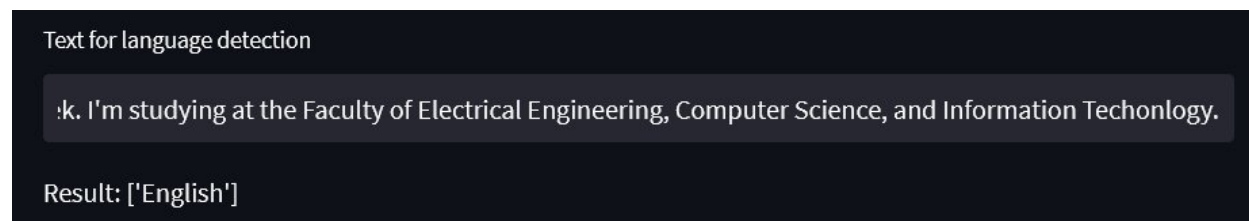
Classification report:				
precision	recall	f1-score	support	
Dutch	0.99	1.00	0.99	215
English	0.82	1.00	0.90	203
Estonian	1.00	0.95	0.98	192
French	0.96	0.99	0.97	180
Indonesian	1.00	0.97	0.98	195
Latin	0.98	0.90	0.94	216
Portugese	1.00	0.96	0.98	204
Romanian	1.00	0.99	1.00	220
Spanish	0.99	0.98	0.99	187
Swedish	1.00	1.00	1.00	184
Turkish	1.00	0.99	0.99	204
accuracy			0.97	2200
macro avg	0.98	0.97	0.97	2200
weighted avg	0.98	0.97	0.97	2200

*Slika 6.1.5. Prikaz metrika Naivnog Bayes algoritma*

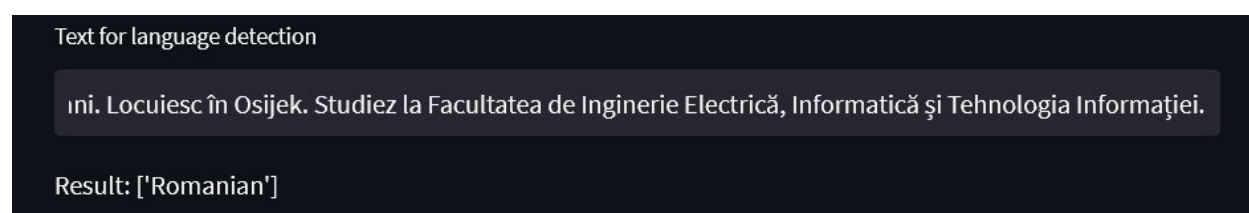
Algoritam je dobro klasificirao sva 3 jezika. Iz matrice zabuna vidimo da je značajnije odstupanje kod latinskog jezika, odnosno vidimo da je algoritam napravio krivu pretpostavku za 16 uzoraka te ih klasificirao kao engleski. Iz metrike sustava vidimo da najmanju preciznost i najmanji odziv ima engleski jezik.



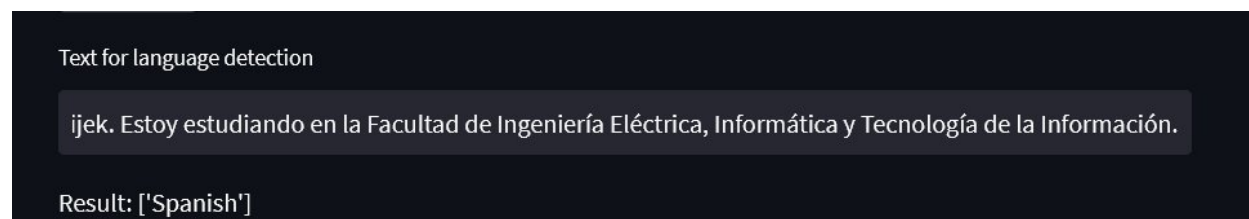
## 6.2. Testiranje i prikaz rezultata logističke regresije



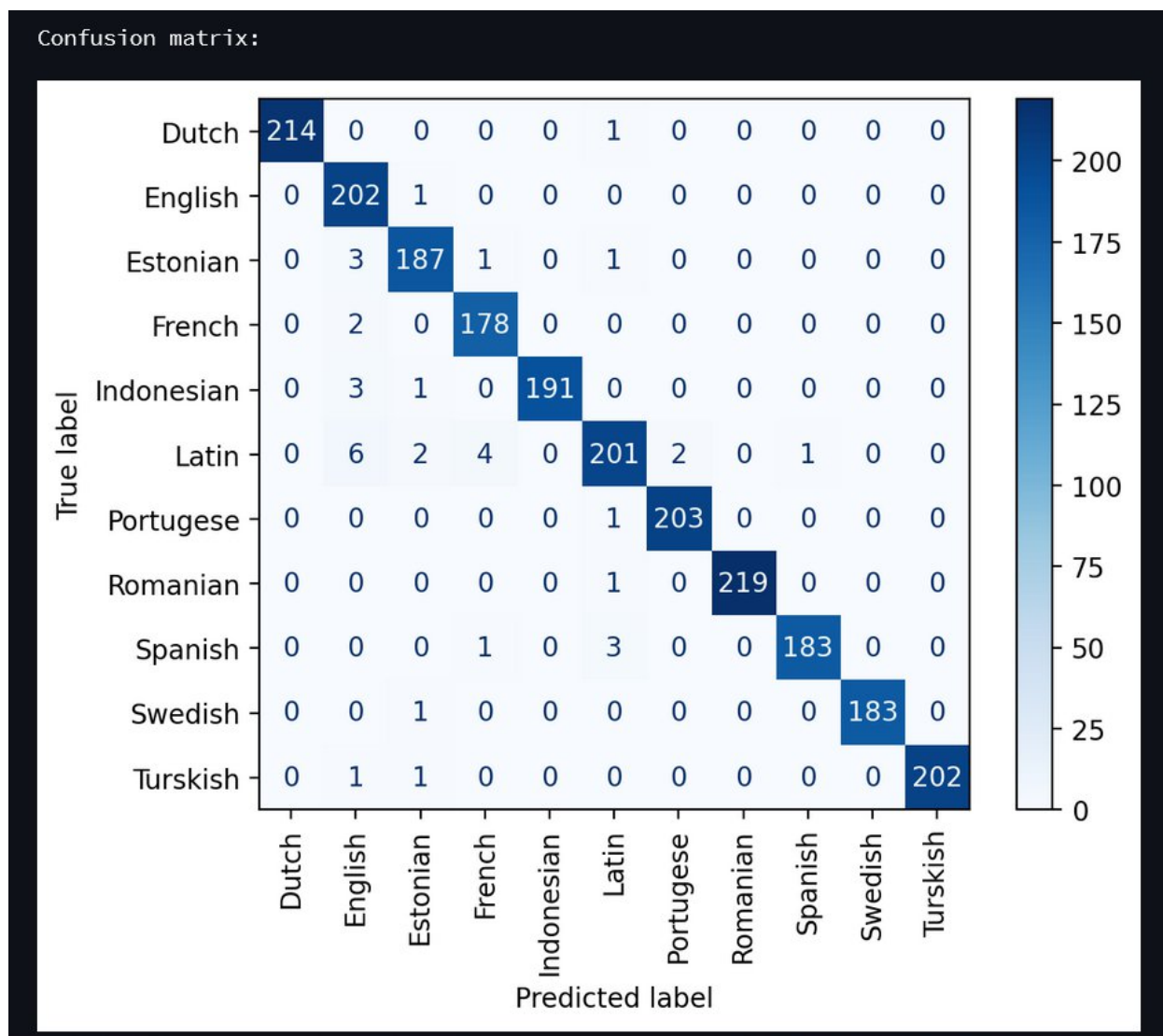
*Slika 6.2.1. Prikaz rezultata Logističke regresije za engleski jezik*



*Slika 6.2.2. Prikaz rezultata Logističke regresije za rumunjski jezik*



*Slika 6.2.3. Prikaz rezultata Logističke regresije za španjolski jezik*



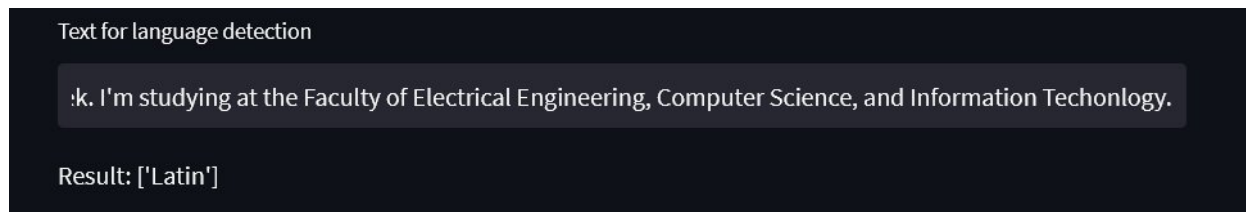
Slika 6.2.4. Prikaz matrice zabune Logističke regresije

Classification report:				
precision	recall	f1-score	support	
Dutch	1.00	1.00	1.00	215
English	0.93	1.00	0.96	203
Estonian	0.97	0.97	0.97	192
French	0.97	0.99	0.98	180
Indonesian	1.00	0.98	0.99	195
Latin	0.97	0.93	0.95	216
Portugese	0.99	1.00	0.99	204
Romanian	1.00	1.00	1.00	220
Spanish	0.99	0.98	0.99	187
Swedish	1.00	0.99	1.00	184
Turkish	1.00	0.99	1.00	204
accuracy			0.98	2200
macro avg	0.98	0.98	0.98	2200
weighted avg	0.98	0.98	0.98	2200

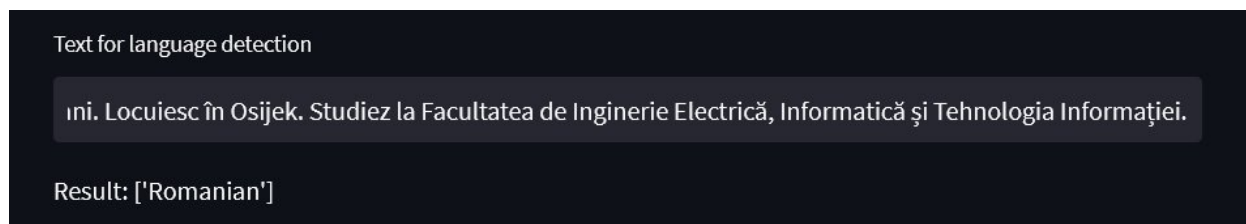
*Slika 6.2.5. Prikaz metrika Logističke regresije*

Vidimo kako je algoritam dobro klasificirao sva 3 jezika. Ponovno najmanju preciznost ima engleski jezik, no ovaj put najmanji odziv i F1 rezultat ima latinski jezik.

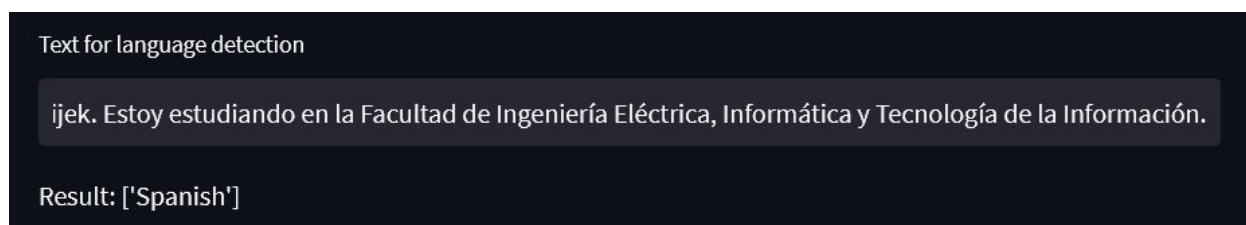
### 6.3. Testiranje i prikaz rezultata K najbližih susjeda



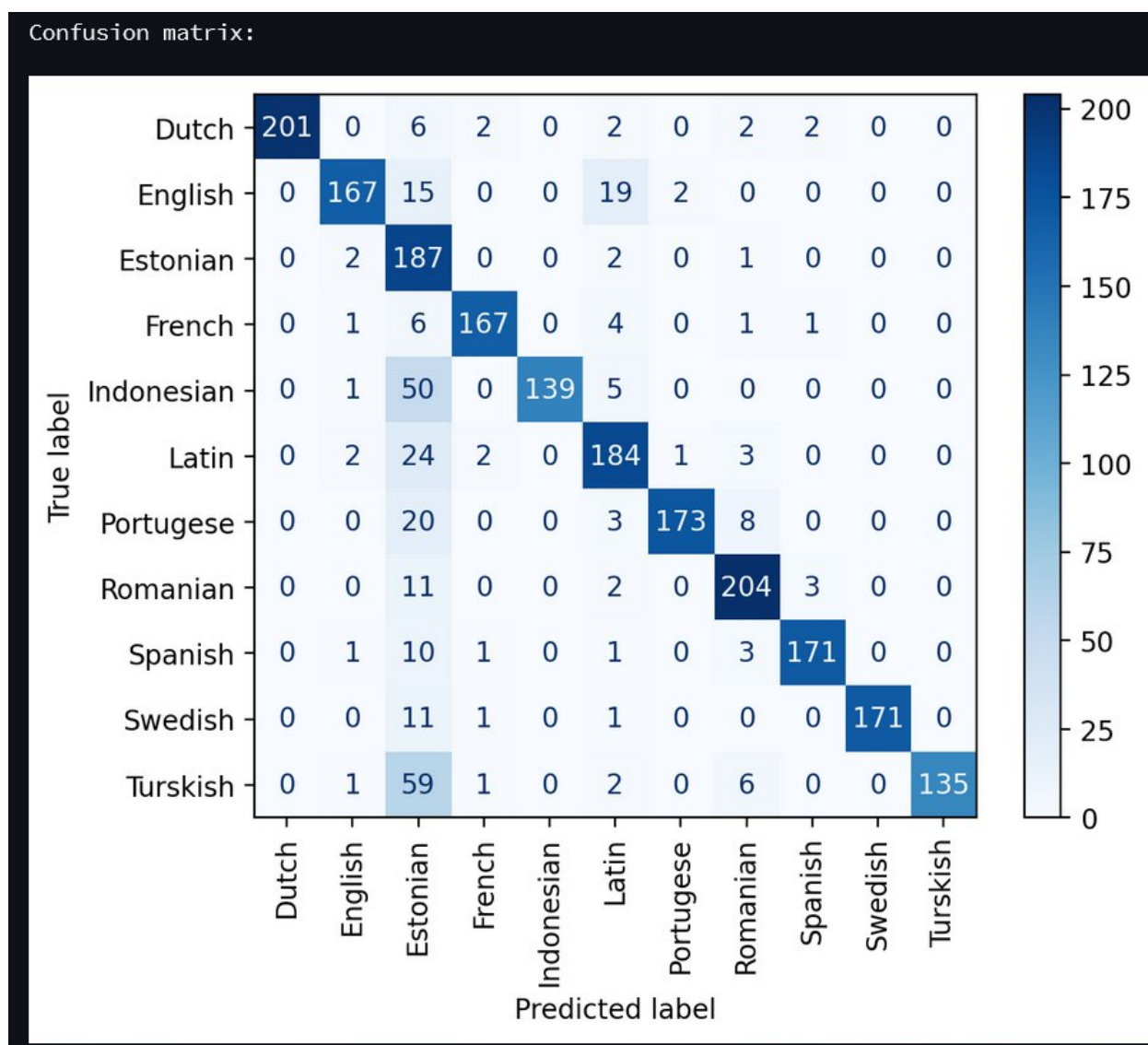
*Slika 6.3.1. Prikaz rezultata K najbližih susjeda za engleski jezik*



*Slika 6.3.2. Prikaz rezultata K najbližih susjeda za rumunjski jezik*



*Slika 6.3.3. Prikaz rezultata K najbližih susjeda za španjolski jezik*



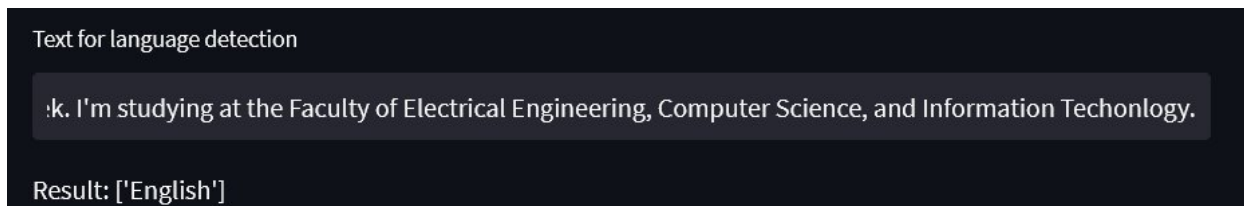
Slika 6.3.4. Prikaz matrice zabune K najbližih susjeda

Classification report:					
precision	recall	f1-score	support		
Dutch	1.00	0.93	0.97	215	
English	0.95	0.82	0.88	203	
Estonian	0.47	0.97	0.63	192	
French	0.96	0.93	0.94	180	
Indonesian	1.00	0.71	0.83	195	
Latin	0.82	0.85	0.83	216	
Portugese	0.98	0.85	0.91	204	
Romanian	0.89	0.93	0.91	220	
Spanish	0.97	0.91	0.94	187	
Swedish	1.00	0.93	0.96	184	
Turkish	1.00	0.66	0.80	204	
accuracy			0.86	2200	
macro avg	0.91	0.86	0.87	2200	
weighted avg	0.91	0.86	0.87	2200	

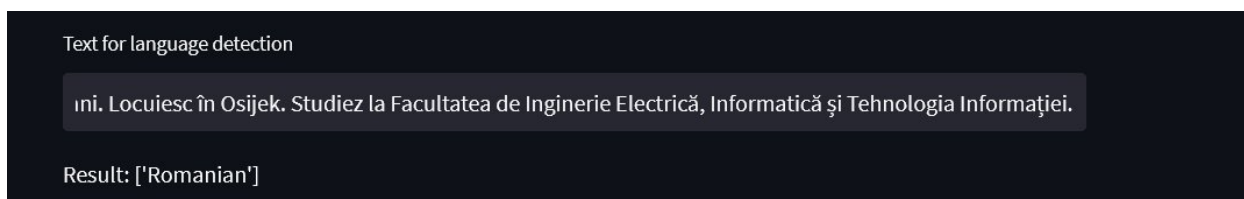
*Slika 6.3.5. Prikaz metrika K najbližih susjeda*

Vidimo da je algoritam krivo klasificirao tekst na engleskom jeziku te da je točnost značajno manja nego u prethodna 2 algoritma. Iz matrice zabune vidimo da je za mnogo uzoraka različitih jezika algoritam klasificirao jezik kao estonski iako nije. Najmanju i najznačajniju razliku preciznosti ima estonski jezik.

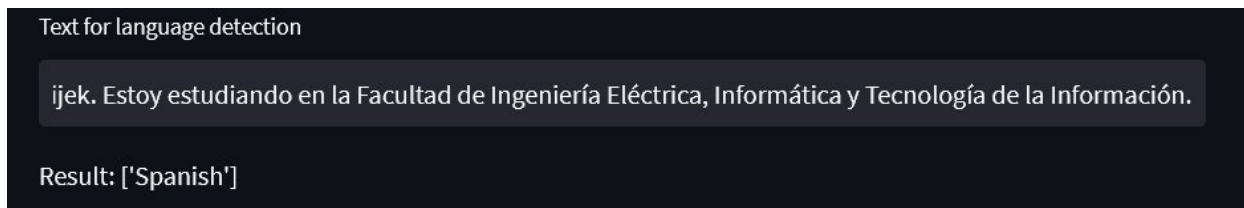
## 6.4. Testiranje i prikaz rezultata strojeva s potpornim vektorima



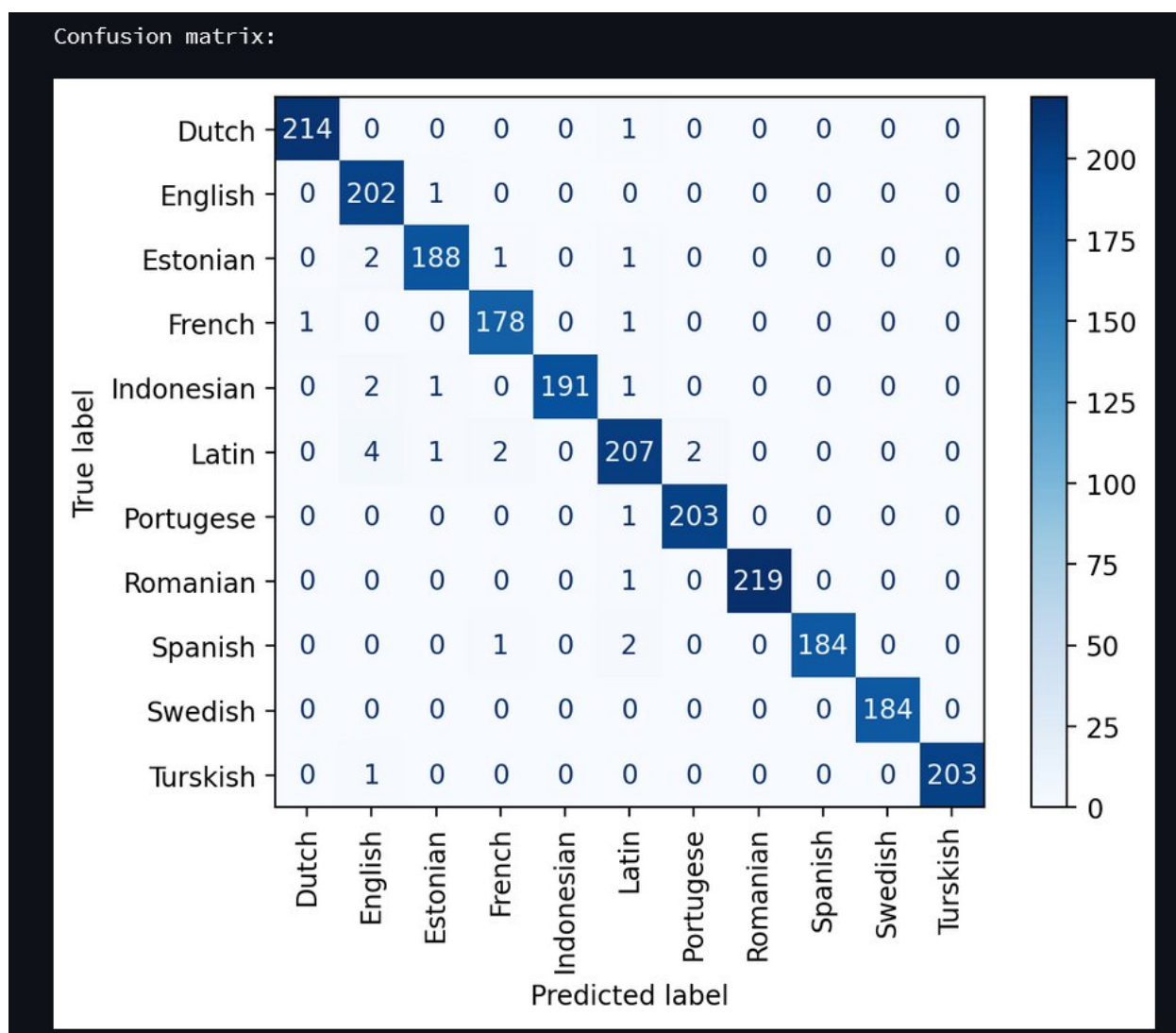
*Slika 6.4.1. Prikaz rezultata strojeva s potpornim vektorima za engleski jezik*



*Slika 6.4.2. Prikaz rezultata strojeva s potpornim vektorima za rumunjski jezik*



*Slika 6.4.3. Prikaz rezultata strojeva s potpornim vektorima za španjolski jezik*



Slika 6.4.4. Prikaz matrice zabune strojeva s potpornim vektorima



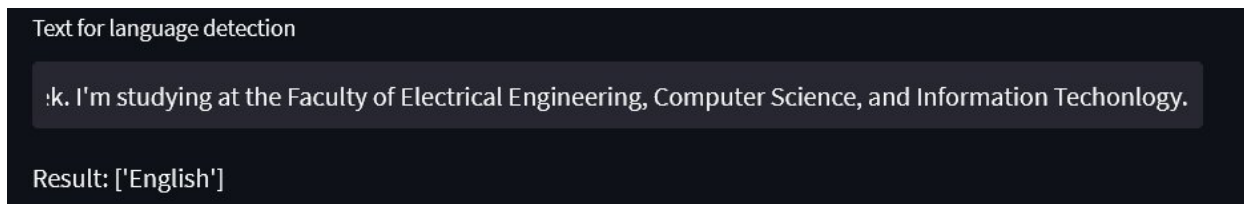
### Classification report:

precision	recall	f1-score	support	
Dutch	1.00	1.00	1.00	215
English	0.96	1.00	0.98	203
Estonian	0.98	0.98	0.98	192
French	0.98	0.99	0.98	180
Indonesian	1.00	0.98	0.99	195
Latin	0.96	0.96	0.96	216
Portugese	0.99	1.00	0.99	204
Romanian	1.00	1.00	1.00	220
Spanish	1.00	0.98	0.99	187
Swedish	1.00	1.00	1.00	184
Turkish	1.00	1.00	1.00	204
accuracy			0.99	2200
macro avg	0.99	0.99	0.99	2200
weighted avg	0.99	0.99	0.99	2200

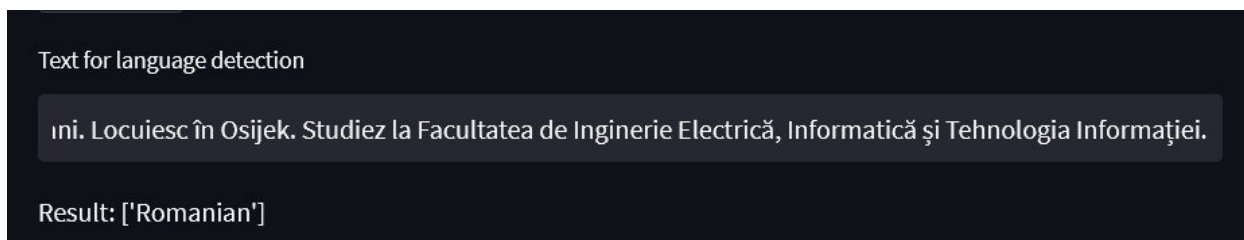
*Slika 6.4.5. Prikaz metrika strojeva s potpornim vektorima*

Ovaj algoritam do sada ima najveće vrijednosti metrika od do sada svih navedenih, sva 3 navedena teksta je dobro klasificirao.

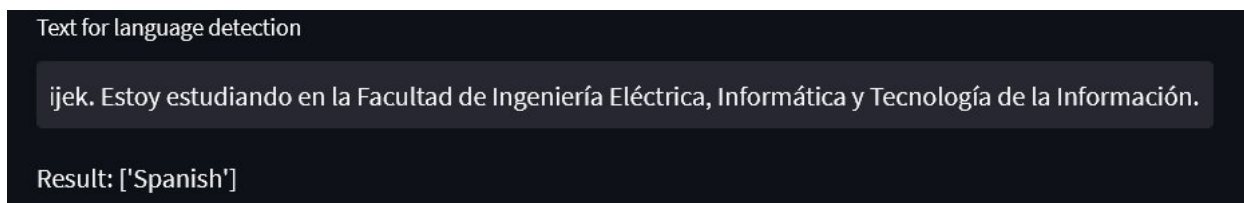
## 6.5. Testiranje i prikaz rezultata algoritma slučajne šume



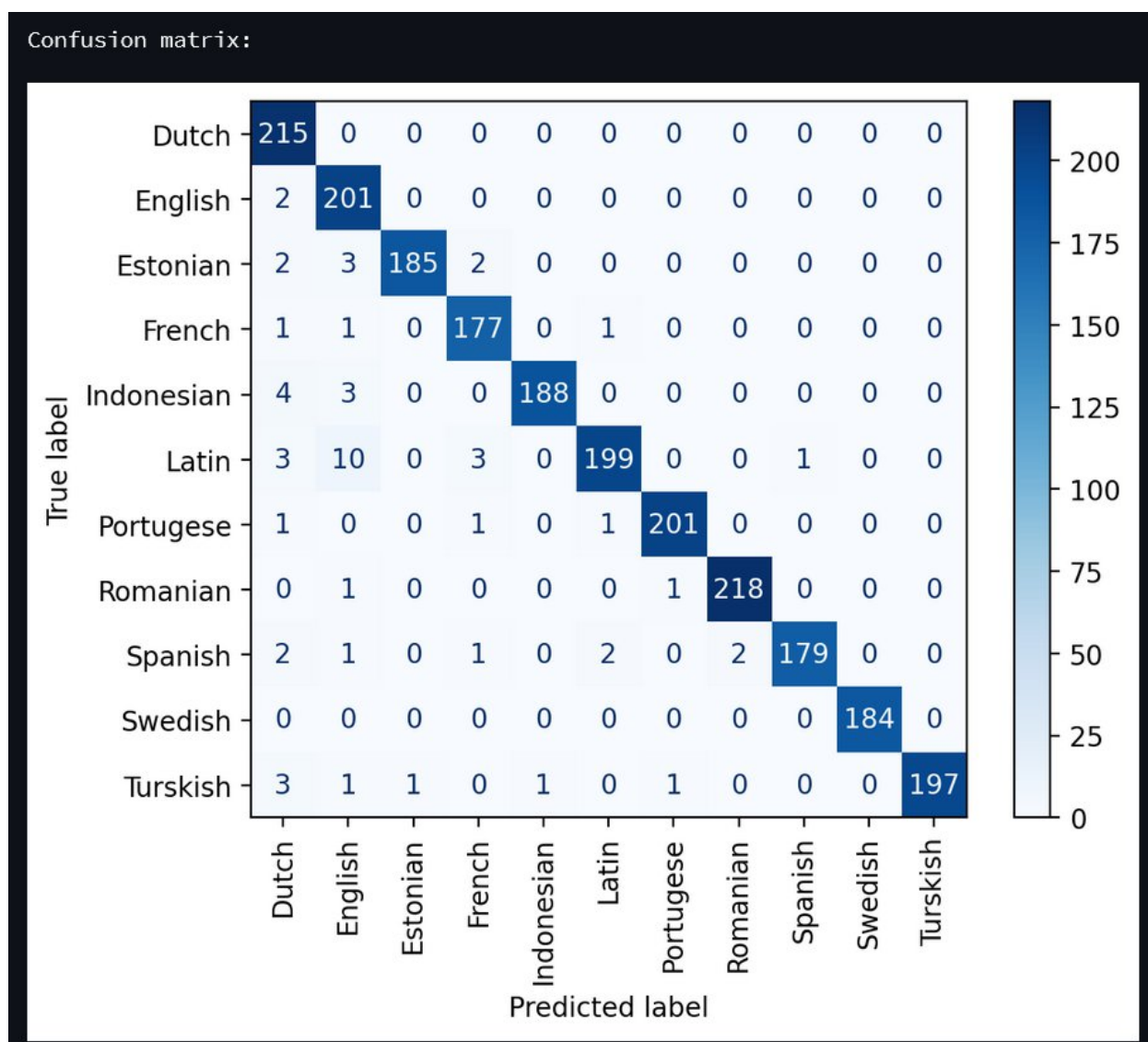
*Slika 6.5.1. Prikaz rezultata slučajne šume za engleski jezik*



*Slika 6.5.2. Prikaz rezultata slučajne šume za rumunjski jezik*



*Slika 6.5.3. Prikaz rezultata slučajne šume za španjolski jezik*

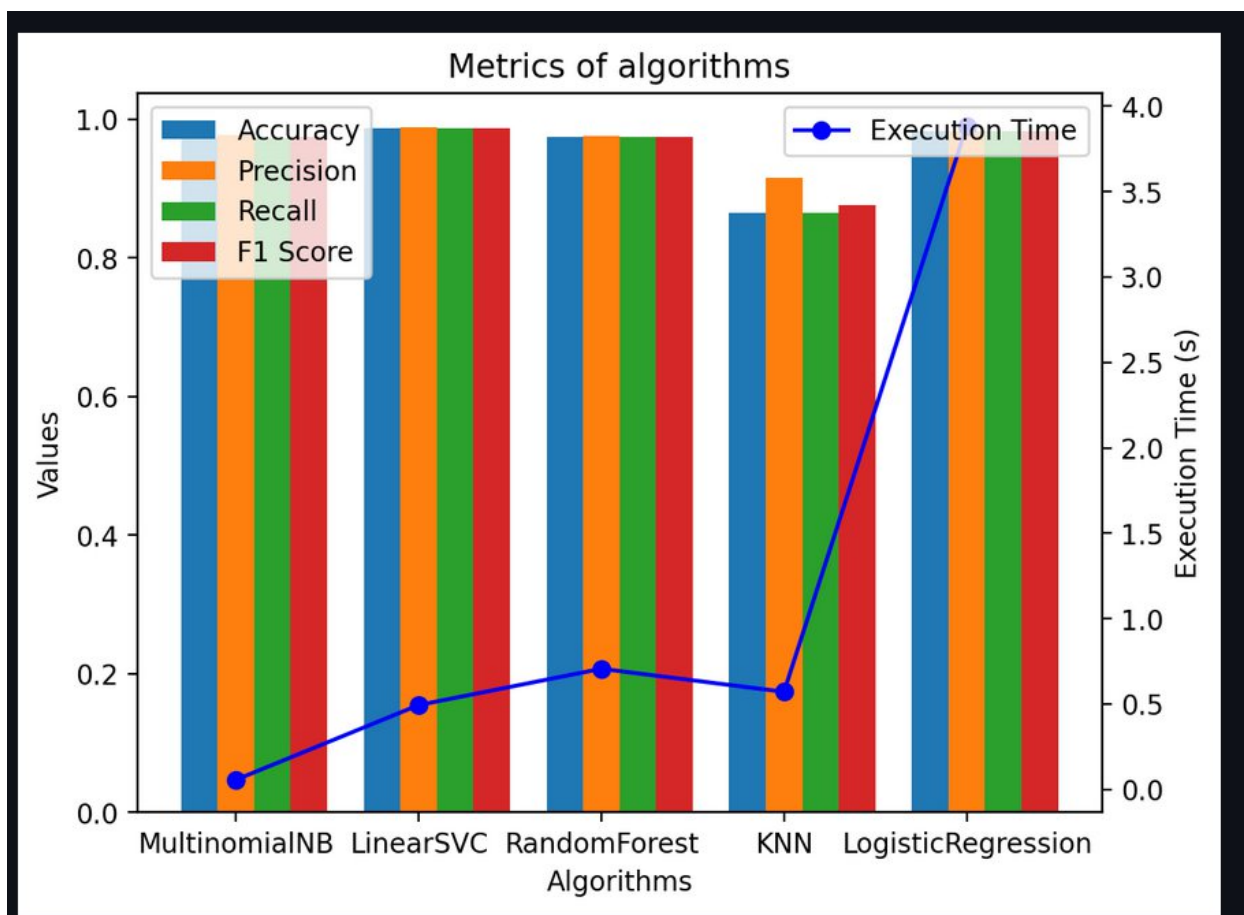


Slika 6.5.4. Prikaz matrice zabune slučajne šume vektorima

Classification report:					
	precision	recall	f1-score	support	
Dutch		0.92	1.00	0.96	215
English		0.91	0.99	0.95	203
Estonian		0.99	0.96	0.98	192
French		0.96	0.98	0.97	180
Indonesian		0.99	0.96	0.98	195
Latin		0.98	0.92	0.95	216
Portuguese		0.99	0.99	0.99	204
Romanian		0.99	0.99	0.99	220
Spanish		0.99	0.96	0.98	187
Swedish		1.00	1.00	1.00	184
Turkish		1.00	0.97	0.98	204
accuracy				0.97	2200
macro avg		0.98	0.97	0.97	2200
weighted avg		0.98	0.97	0.97	2200

*Slika 6.5.5. Prikaz metrika slučajne šume*

I ovaj je algoritam također dobro klasificirao sva 3 navedena teksta, ima nešto lošije rezultate od logističke regresije i SVM algoritma, ali i dalje ima visoke vrijednosti metrika evaluacije.



Slika 6.5.5. Prikaz metrika svih i vrijeme izvođenja svih 5 algoritama

## 7. ZAKLJUČAK

Ovim projektom su se testirali različiti algoritmi strojnog učenja za detekciju jezika nekog teksta. Svi algoritmi su uglavnom dobro klasificirali jezik napisanog teksta. Najbolji algoritam prema metrikama evaluacije je SVM. Najlošiji algoritam prema metrikama evaluacije je K najbližih susjeda. Njegova točnost je iznosila 86%. Također je pri testiranju krivo klasificirao engleski jezik kao latinski. Najbrži algoritam pokazao se Naivni Bayes algoritam. Najsporiji algoritam je Logistička regresija. Isto tako veća je vjerojatno da će algoritmi dobro klasificirati jezik teksta što je tekst duži.

## 8. LITERATURA

- [1] Dataset: <https://www.kaggle.com/code/nehahatti/language-detection-project/input>
- [2] Streamlit: <https://docs.streamlit.io/>
- [3] scikit-learn: <https://scikit-learn.org/stable/>
- [4] Anaconda: [https://en.wikipedia.org/wiki/Anaconda\\_\(Python\\_distribution\)](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution))