Matematički fakultet

Univerzitet u Beogradu

Seminarski rad:

**Uticaj P-adičnosti na razilke genetskog koda SARS1, SARS2 i MERS koronavirusa**

Mentor: Studenti:

Prof. dr Nenad Mitić Marko Koprivica 61/2021 Milan Bodo 173/2021

[**1.Uvod 3**](#_gjdgxs)

[**2.P-adična norma 3**](#_30j0zll)

[2.1 P-adično rastojanje 3](#_1fob9te)

[2.2 Primena p‑adičnih metrika u analizi genetskog koda 4](#_nu5oawy2li4l)

[2.3 P-adično rastojanje izmedju kodona primenjeno u našem istraživanju 5](#_2et92p0)

[**3.Podaci 5**](#_tyjcwt)

[3.1 Pretprocesiranje podataka 7](#_3dy6vkm)

[3.2 Podela podataka 8](#_1t3h5sf)

[**4. Modeli 8**](#_4d34og8)

[4.1 Modeli klasifikacije po tipu virusa 8](#_2s8eyo1)

[4.1.1 Model 1 9](#_17dp8vu)

[4.1.2 Model 2 11](#_3rdcrjn)

[4.1.3 Model 3 12](#_26in1rg)

[4.1.4 Model 4 14](#_lnxbz9)

[4.2 Modeli klasifikacije po tipu proteina 15](#_35nkun2)

[4.2.1 Model 5 15](#_1ksv4uv)

[4.2.2 Model 6 17](#_44sinio)

[4.2.3 Model 7 19](#_2jxsxqh)

[4.2.4 Model 8 21](#_z337ya)

[4.3 Modeli klasifikacije po soju SARS2 virusa 23](#_3j2qqm3)

[4.3.1 Model 9 23](#_1y810tw)

[4.3.2 Model 10 25](#_4i7ojhp)

[4.3.3 Model 11 27](#_1ci93xb)

[4.3.4 Model 12 29](#_2bn6wsx)

[4.4 Unakrsna provera modela 32](#_qsh70q)

[**5. Zaključak 34**](#_1pxezwc)

[**6.Reference 34**](#_49x2ik5)

# 1.Uvod

Genetski kod predstavlja temelj života, čuvajući informacije neophodne za sintezu proteina u svim organizmima. Razumevanje strukture i razlika u genetskim sekvencama virusa od ključnog je značaja za unapređenje dijagnostike, lečenja i prevencije zaraznih bolesti. Ovaj seminarski rad istražuje uticaj p-adičnosti, matematičkog koncepta zasnovanog na teoriji brojeva, na razlike u genetskom kodu tri značajna koronavirusa: SARS1, SARS2 i MERS.

P-adična analiza omogućava inovativan pristup istraživanju genetskih sekvenci, koristeći ultrametričku distancu za kvantifikaciju sličnosti između nukleotidnih sekvenci. U radu se detaljno opisuje kako se p-adična metrika primenjuje na nukleotidne sekvence, uzimajući u obzir razlike na nivou pojedinačnih kodona. Posebna pažnja posvećena je implementaciji algoritama za klasifikaciju virusa i proteina, kao i analizi rezultata postignutih korišćenjem različitih modela mašinskog učenja.

Rad se oslanja na bogat skup podataka koji obuhvata sekvence proteina različitih sojeva SARS2 virusa, kao i sekvence SARS1 i MERS virusa. Istraživački fokus je na proceni tačnosti modela klasifikacije koji koriste p-adičnu metriku u poređenju sa jednostavnijim metrikama. Rezultati ukazuju na značajan uticaj p-adičnosti u prepoznavanju razlika među virusima i njihovim proteinima.

Kroz detaljnu analizu modela, uključujući algoritme K-najbližih suseda, rad pruža uvid u efikasnost p-adične metrike u kontekstu bioloških podataka. Ovi nalazi doprinose razumevanju potencijala p-adičnih metoda u biomedicinskim istraživanjima i otvaraju nove mogućnosti za primenu u analizi genetskih podataka.

# 2.P-adična norma

Kako se ∀m∈Z , m≠0 može zapisati kao m=pk⋅a gde je p prost broj, k ∈{0,1,2,...} i a ceo broj koji nije delji sa p. P-adična norma broja m u oznaci |m|p  se definiše na sledeći način:

|m|p = p-k [1]

i specijalno za m=0, |m|p=0.

## 2.1 P-adično rastojanje

P-adično rastojanje dva cela broja x i y se računa na sledeći način:

|x-y|p

P-adična metrika je ultrametrika, to jest za nju važi:

1. |x-y|p ≥ 0 , |x-y|p＝0 ⇔x＝y

2. |x-y|p＝|y-x|p

3. |x-y|p ≤ max {|x-z|p ,|y-z|p}

Primeri računja p-adičnog rastojanja:

|63-3|2= |60|2= |22⋅3⋅5|2= 1/4

|63-3|3= |60|3= |22⋅3⋅5|3=⅓

## 2.2 Primena p‑adičnih metrika u analizi genetskog koda

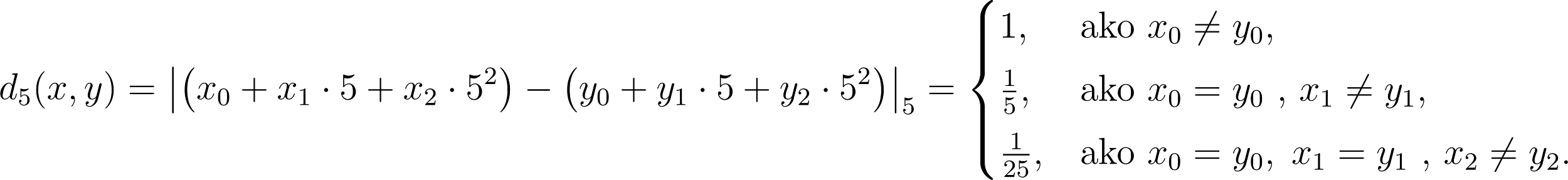
Predstavljanje genetskog koda u okviru p‑adičnih metrika omogućava merenje sličnosti između kodona. U ovom pristupu, kodoni se modeliraju kao trocifreni brojevi u 5‑adičnom sistemu, pri čemu svaka cifra odgovara jednoj nukleotidnoj bazi. Moguće je uspostaviti sledeću numeričku dodelu:

* Citozin (C) = 1
* Adenin (A) = 2
* Timidin (T) = 3
* Guanin (G) = 4

Na taj način, svaki kodon x=x0x1x2 kodiran u 5-adičnom sistemumože se predstaviti brojem

x=x0+x1⋅5+x2⋅52.

P-adično rastojanje između dva kodona x=x0x1x2 i y=y0y1y2, za p=0​ definiše se na sledeći način:

[](https://www.codecogs.com/eqnedit.php?latex=d_5(x%2Cy)%20%3D%20%5Cleft%7C%5Cleft(x_0%20%2B%20x_1%5Ccdot5%20%2B%20x_2%5Ccdot5%5E2%5Cright)%20-%20%5Cleft(y_0%20%2B%20y_1%5Ccdot5%20%2B%20y_2%5Ccdot5%5E2%5Cright)%5Cright%7C_5%20%3D%20%5Cbegin%7Bcases%7D%201%2C%20%26%20%5Ctext%7Bako%20%7D%20x_0%20%5Cneq%20y_0%2C%5C%5C%5B1mm%5D%20%5Cfrac%7B1%7D%7B5%7D%2C%20%26%20%5Ctext%7Bako%20%7D%20x_0%20%3D%20y_0%20%5Ctext%7B%20%2C%20%7D%20x_1%20%5Cneq%20y_1%2C%5C%5C%5B1mm%5D%20%5Cfrac%7B1%7D%7B25%7D%2C%20%26%20%5Ctext%7Bako%20%7D%20x_0%20%3D%20y_0%2C%5C%3B%20x_1%20%3D%20y_1%20%5Ctext%7B%20%2C%20%7D%20x_2%20%5Cneq%20y_2.%20%5Cend%7Bcases%7D%20#0)

**Slika 1** Izračunavanje 5-adičnog rastojanja izmedju dva kodona [1]

Ovaj izraz prikazuje da su kodoni najbliži (tj. d5(x,y)=1/25 je minimalno​) ukoliko se razlikuju isključivo na trećoj poziciji, dok se u slučaju razlikovanja na prvoj poziciji dobija maksimalna vrednost rastojanja d5(x,y)=1.

Pored 5‑adične metrike, primena 2‑adične metrike dodatno odredjuje sličnost među nukleotidima, naročito u kontekstu razlikovanja purina (A i G) i pirimidina (C i T). Kombinovanjem ove dve metrika, moguće je izgraditi hijerarhijsku strukturu kodonskog prostora koja direktno korelira sa biološkom funkcijom degeneracije genetskog koda: kodoni koji su najbliži prema 5‑adičnoj i 2‑adičnoj metrici češće kodiraju istu aminokiselinu ili signal za prekid sinteze proteina.

**Primer izračunavanja 5‑adičnog rastojanja između kodona**

Razmotrimo dva kodona:

* x=ACA
* y=AAA

Koristeći gore navedenu dodelu (C = 1, A = 2, T = 3, G = 4), dobijamo:

* Za kodon x:  
  x0=A=2,   x1=C=1,  x2=A=2
* Za kodon y:  
  y0=A=2,  y1=A=2,  y2=A=2.

Sada izračunajmo 5-adično rastojanje ovih kodona (primenimo postupak sa slike 1).

[](https://www.codecogs.com/eqnedit.php?latex=d_5(x%2Cy)%20%3D%20%5Cleft%7C%5Cleft(2%20%2B%201%5Ccdot5%20%2B%202%5Ccdot5%5E2%5Cright)%20-%20%5Cleft(2%2B%202%5Ccdot5%20%2B%202%5Ccdot5%5E2%5Cright)%5Cright%7C_5%20%3D%20%5Cleft%7C57%20-62%5Cright%7C_5%20%3D%5Cleft%7C-5%5Cright%7C_5%3D1%2F5%20#0)

**Slika 2** primer izračunavanja 5-adičnog rastojanja

Alternativno, posmatramo cifre (ili nukleotide). Budući da su prve cifre kodona x i y jednake, posmatramo druge dve, primetimo da se one razlikuju, što po definiciji sa slike 1 znači da je rastojanje ova dva kodona 1/5.

## 2.3 P-adično rastojanje izmedju kodona primenjeno u ovom istraživanju

Za izračunavanje distance izmedju kodona korišćena je kombinacija 5-adične i 2-adične metrike i to na sledeći način. Ako se kodoni razlikuju na prvoj nukleotidi, rastojanje se uvećava za 50=1, ako se kodoni razlikuju na drugoj nukleotidi, rastojanje se uvećava za 5-1=1/5, dok ako se kodoni razlikuju na trećoj nukleotidi i apsolutna razlika kodova na toj poziciji je 2 (što znači da su obe nukleotide na trećoj poziciji ili purinske ili pirinske) distnaca se uvećava za 2-1⋅5-2=1/2⋅1/25 ili ako ta razlika nije 2 rastojanje se uvećava za 20⋅5-2=1⋅1/25. Rastojanje⋅ izmedju sekvenci predstavlja zbir distanci izmedju kodona.

# 3.Podaci

Istraživanje je sprovedeno na bazi podataka koja sadrži 13,202 instance. Ključni atributi uključuju ime virusa, naziv proteina i odgovarajuću nukleotidnu sekvencu. Podaci su preuzeti sa zvanične baze [2], pri čemu su sekvence proteina odabrane tako da budu nukleotidno kompletne i bez višeznačnih karaktera. Nakon toga, primenjena je restrikcija kako bi se obezbedila jedinstvenost sekvenci proteina, datoteka sa podacima se nalazi na putanji *data/sars2\_mers\_sars1.txt.* U nastavku se nalaze tabele raspodele podataka po tipu virusa, tipu proteina i soju SARS2 virusa po WHO klasifikaciji.

| Virus | Broj instanci |
| --- | --- |
| SARS1 | 18 |
| MERS | 1859 |
| SARS2 | 11325 |

**Tabela 1** Raspodela instanci po tipu virusa

| Soj SARS2 virusa | Broj instanci |
| --- | --- |
| Alpha, Delta, Epsilon, Gamma, Iota, Omricon | po 1500 |
| Eta | 628 |
| Beta | 586 |
| Zeta | 467 |
| Lambda | 410 |
| Kappa | 157 |
| Mu | 65 |
| Theta | 20 |

**Tabela 2** Raspodela instanci po soju SARS2 virusa po WHO klasifikaciji

| Protein | Broj instanci |
| --- | --- |
| ORF1ab polyprotein | 5755 |
| ORF1a polyprotein | 4153 |
| surface glycoprotein | 1666 |
| nucleocapsid phosphoprotein | 640 |
| ORF3a protein | 221 |
| membrane glycoprotein | 136 |
| ORF4b protein | 128 |
| ORF3 protein | 104 |
| ORF5 protein | 104 |
| ORF8 protein | 76 |
| ORF4a protein | 67 |
| ORF8b protein | 59 |
| envelope protein | 40 |
| ORF7a protein | 25 |
| ORF6 protein | 10 |
| ORF7b protein | 9 |
| ORF1b polyprotein | 7 |
| ORF10 protein | 2 |

**Tabela 3** Raspodela instanci po tipu proteina

## 3.1 Pretprocesiranje podataka

Za izračunavanje rastojanja između sekvenci koristi se program implementiran u C++, čiji se izvorni kod nalazi u datoteci *source/create\_distances\_triangle.cpp*. Ovaj program generiše tekstualnu datoteku koja sadrži trougaonu matricu sa izračunatim vrednostima rastojanja između sekvenci.

Radi optimizacije skladištenja i efikasnijeg pristupa podacima, matrica se serijalizuje i kompresuje pomoću funkcije serialize\_and\_compress \_distance\_matrix, koja se nalazi u skripti *source/fileDistanceProcessing.py*. Ovoj funkciji se prosleđuju:

1. putanja do tekstualne datoteke sa prethodno generisanom matricom rastojanja,
2. putanja do rezultujuće datoteke u kojoj će biti sačuvani serijalizovani i kompresovani podaci.

Za rekonstrukciju originalne matrice koristi se funkcija deserialize\_and\_decompress \_distance\_matrix, koja prima putanju do kompresovane datoteke i vraća deserijalizovanu matricu rastojanja. U funkcijama za serijalizaciju i deserijalizaciju korišćena je biblioteka pickle [3].

Ovaj pristup značajno poboljšava efikasnost prikaza i pristupa podacima o rastojanju između svih parova sekvenci, optimizujući proces analize i daljih istraživanja.

Datoteka *source/create\_distances\_triangle.cpp* sadrži implementaciju četiri različite funkcije za izračunavanje matrice rastojanja između sekvenci. Ove funkcije koriste dve različite metrike:

1. **P-adično rastojanje,**
2. **Hamingovo rastojanje**

Pored različitih metrika, funkcije se razlikuju i po načinu odsecanja sekvenci, što rezultira sledećim pristupima:

1. **Fiksno odsecanje na dužinu najkraće sekvence** – sve sekvence se skraćuju na dužinu najkraće sekvence u skupu podataka (u ovom slučaju 78 kodona).
2. **Adaptivno odsecanje u zavisnosti od poređenih sekvenci** – pri svakom računanju rastojanja između dve sekvence, sekvence se skraćuju na dužinu kraće od njih.

Ove četiri metode generišu četiri različite matrice rastojanja, koje su sačuvane u sledećim datotekama:

* *data/hamming\_distances\_full.zip* – matrica Hamingovog rastojanja sa fiksnim odsecanjem,
* *data/hamming\_distances\_clipped.zip* – matrica Hamingova rastojanja sa adaptivnim odsecanjem,
* *data/padic\_distances\_full.zip* – matrica P-adičnog rastojanja sa fiksnim odsecanjem,
* *data/padic\_distances\_clipped.zip* – matrica P-adičnog rastojanja sa adaptivnim odsecanjem.

## 3.2 Podela podataka

Podaci su podeljeni na trening i test skup u odnosu 2:1 (trening : test) primenom stratifikovanog uzorkovanja prema ciljnoj klasi. Stratifikacija je izvršena u skladu sa:

* vrstom virusa,
* vrstom proteina,
* sojem SARS2 virusa prema WHO klasifikaciji.

Ovim pristupom napravljene su tri različite podele podataka. Za podelu podataka je korišćena funkcija train\_test\_split iz scikit-learn biblioteke [4]. Tako podeljeni podaci nalaze se u train.csv i test.csv datotekama u direktorijumima *data/virus, data/proteins* i *data/sars2\_who*, pored .csv datoteka ovde se nalaze i podaci u formatu pogodnom za korišćene u ovom istraživanju, a to su samo redni brojevi sekvenci u matricama rastojanja i njihove klase .

Ovi podaci mogu se jednostavno učitati pomoću odgovarajućih funkcija iz skripte *source/fileDistanceProcessing.py*:

* train\_test\_virus – za podelu po vrstama virusa,
* train\_test\_protein – za podelu po vrstama proteina,
* train\_test\_sars2 – za podelu prema WHO klasifikaciji SARS-CoV-2 sojeva.

Ovakva organizacija podataka omogućava efikasno rukovanje različitim klasifikacionim problemima unutar istraživanja.

# 4. Modeli

Za trening i prikaz rezultata modela koji koriste algoritam K-najbližih suseda koristi se funkcija perform\_and\_evaluate\_model,koja pokreće trening modela, ispisuje matricu konfuzije i tačnost na trening i test skupu.

## 4.1 Modeli klasifikacije po tipu virusa

U narednim podpoglavljima biće prikazani modeli klasifikacije po tipu virusa, koristeći različlite algoritme i rastojanja između sekvenci.

### 4.1.1 Model 1

Ovaj model primenjuje algoritam K-najbližih suseda(*KNeighborsClassifier*) iz biblioteke *scikit-learn* za klasifikaciju tipova virusa. Model koristi **P-adičnu metriku** za izračunavanje rastojanja između sekvenci, uz primenu **adaptivnog odsecanja**. Parametar broja suseda postavljen je na **3**.

Implementacija modela nalazi se u Jupyter svesci *source/KNN.ipynb* pod naslovom Model 1.

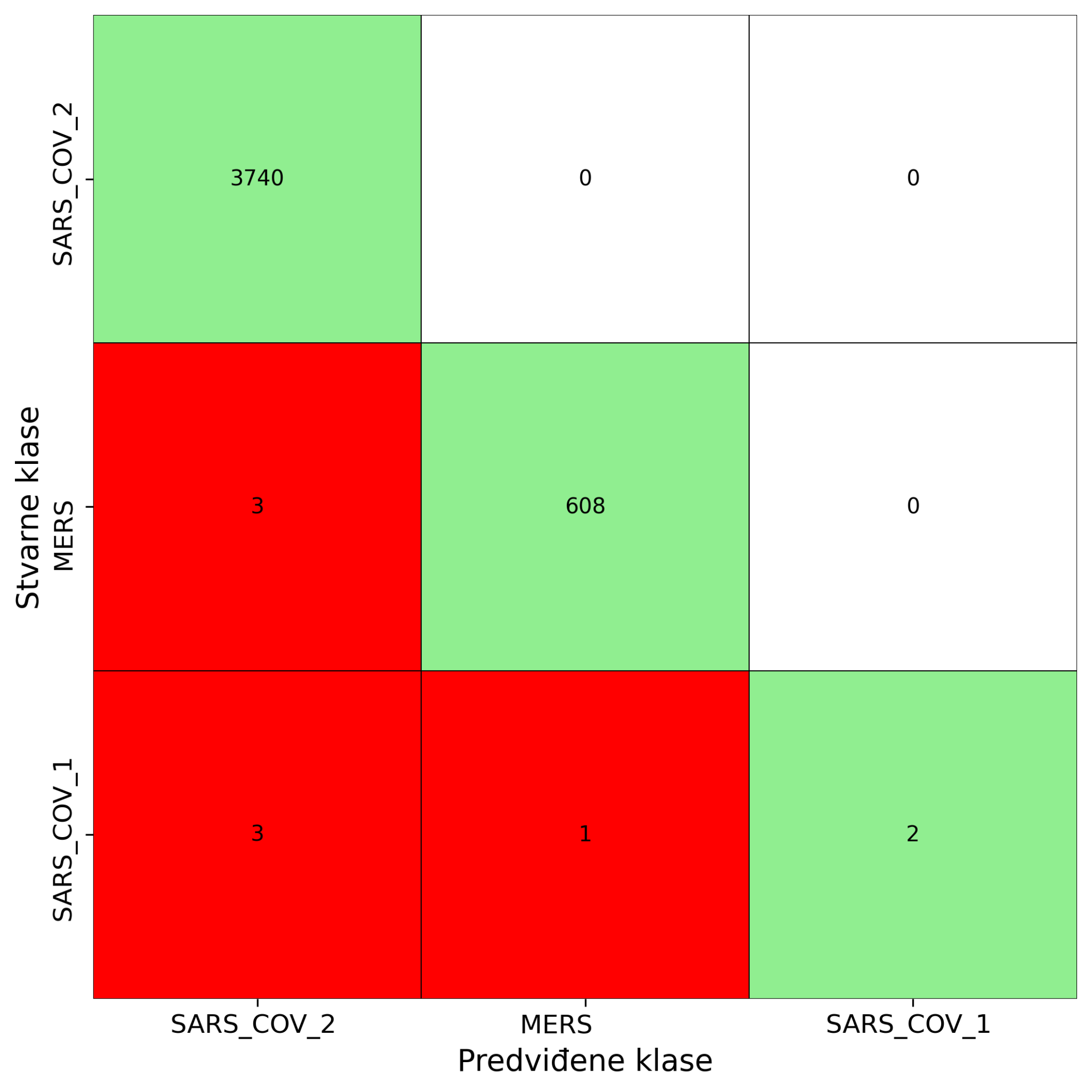
**Ocena modela:**

Dobijene su sledeće vrednosti tačnosti modela na trening i test skupu:

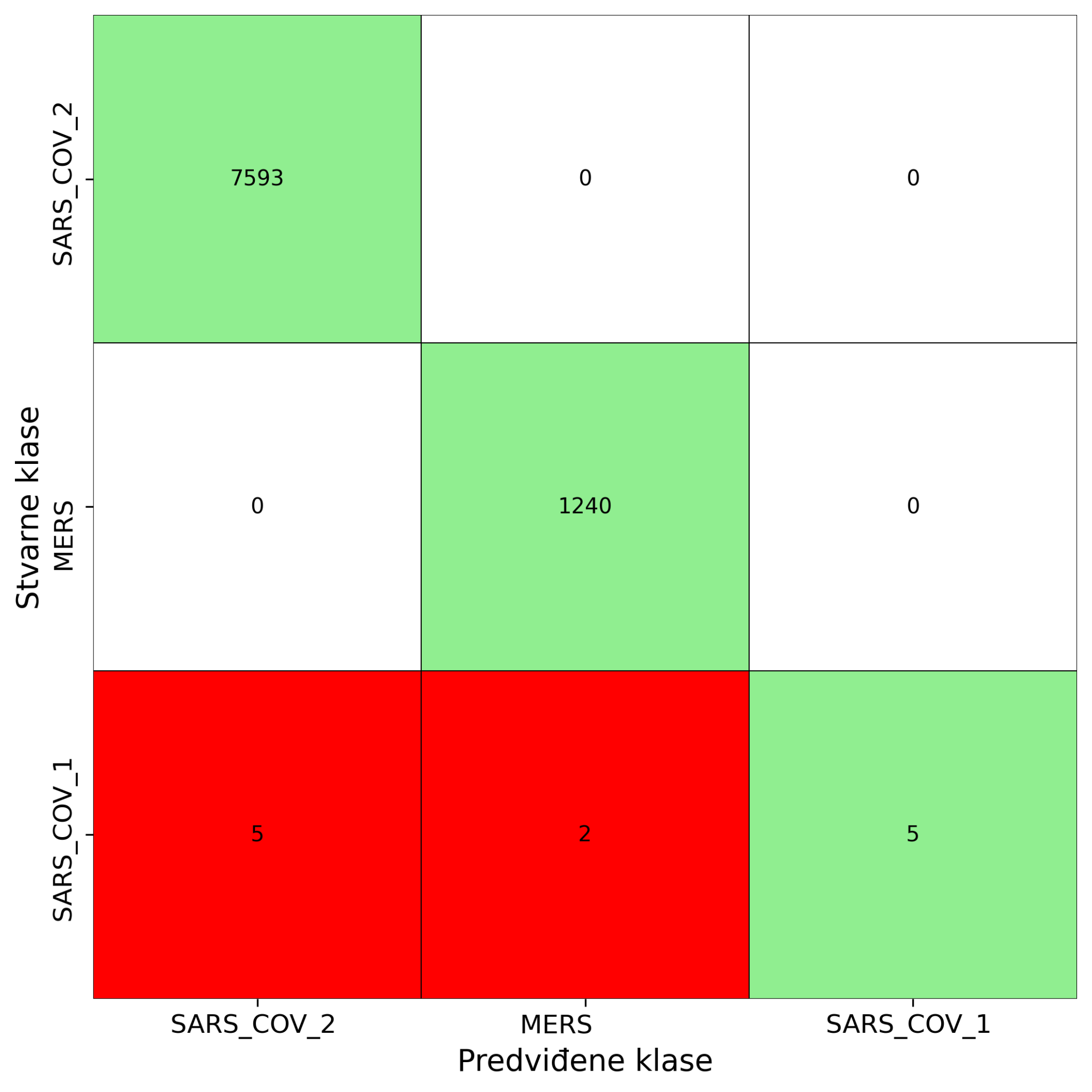
* **Trening skup:** 0.999
* **Test skup:** 0.998

U nastavku su prikazane matrice kofuzije Modela 1 na trening i test skupu, takodje ove matrice se nalaze i na putanjama *results/confusion\_matrix/train\_padic\_full\_virus.csv* i *results/confusion\_matrix/test\_padic\_full\_virus.csv*.

Na dijagonali matrice konfuzije, koja je označena zelenom bojom, prikazana su tačna predviđanja modela, odnosno slučajevi u kojima je model ispravno klasifikovao instance. Crvenom bojom su označeni pogrešni rezultati, odnosno slučajevi u kojima je model pogrešio u klasifikaciji. Bela boja označava kombinacije klasa za koje model nije pravio greške, odnosno gde nije bilo predviđanja. Funkcija u kojoj se definiše izgled matrice konfuzije zove se visualize\_confusion\_matrix i nalazi se u datoteci na putanji *source/KNN.ipynb*, za iscrtavanje matrice konfuzije korišćena je funkcija heatmap iz biliboteke seaborn[5]. Ovakav način prikaza matrice konfuzije će koristiti u prikazima matrica konfuzije u celom radu.

****

**Tabela 4** Matrica konfuzije Modela 1 na test skupu

****

**Tabela 5** Matrica konfuzije Modela 1 na trening skupu

### 4.1.2 Model 2

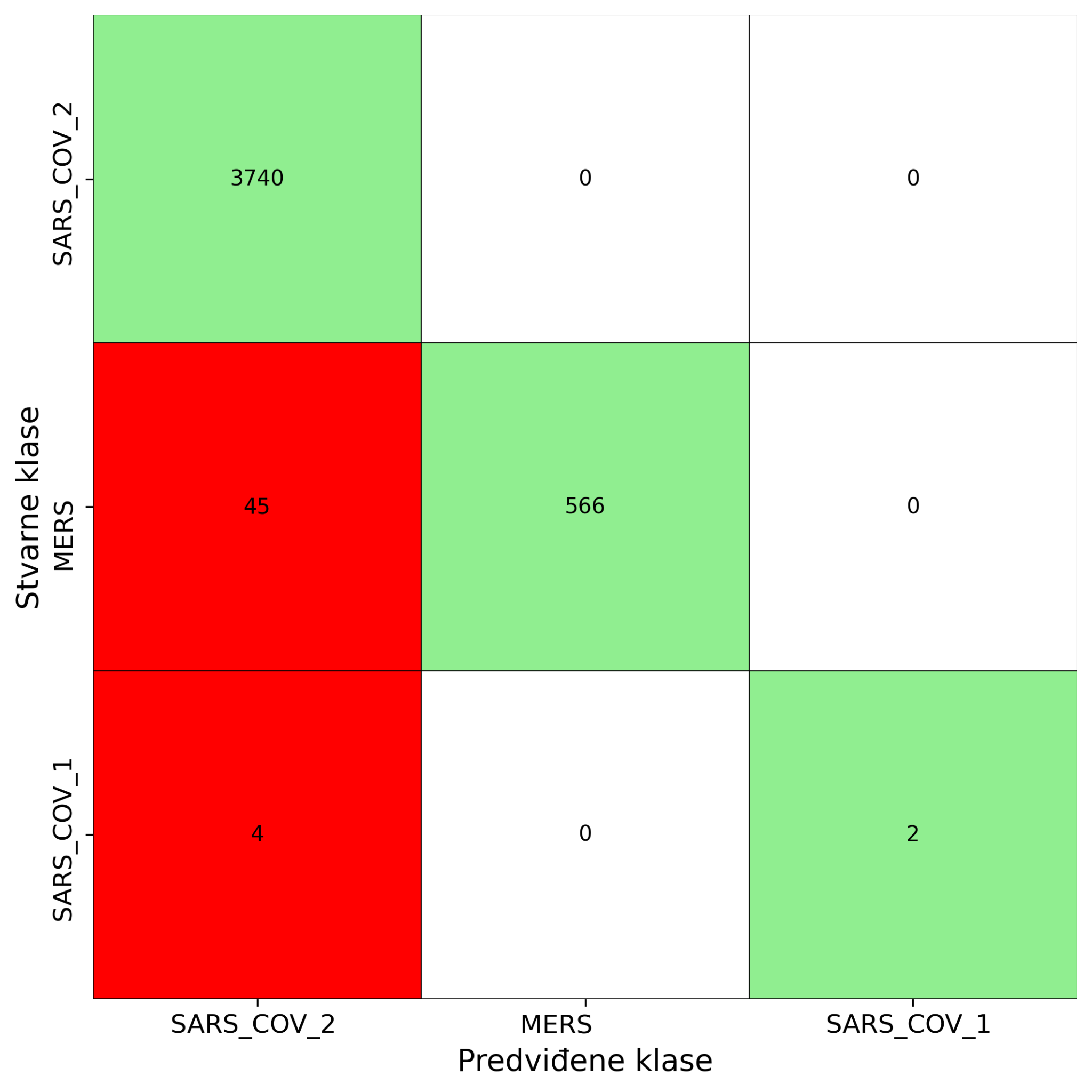
Ovaj model koristi algoritam K-najbližih suseda (*KNeighborsClassifier*) iz biblioteke *scikit-learn* za klasifikaciju tipova virusa. Za izračunavanje rastojanja između sekvenci primenjuje se **Hamingovo rastojanje**, uz **adaptivno odsecanje sekvenci**. Broj suseda u modelu postavljen je na **3**.

Implementacija modela nalazi se u Jupyter svesci *source/KNN.ipynb* pod naslovom **Model 2**.

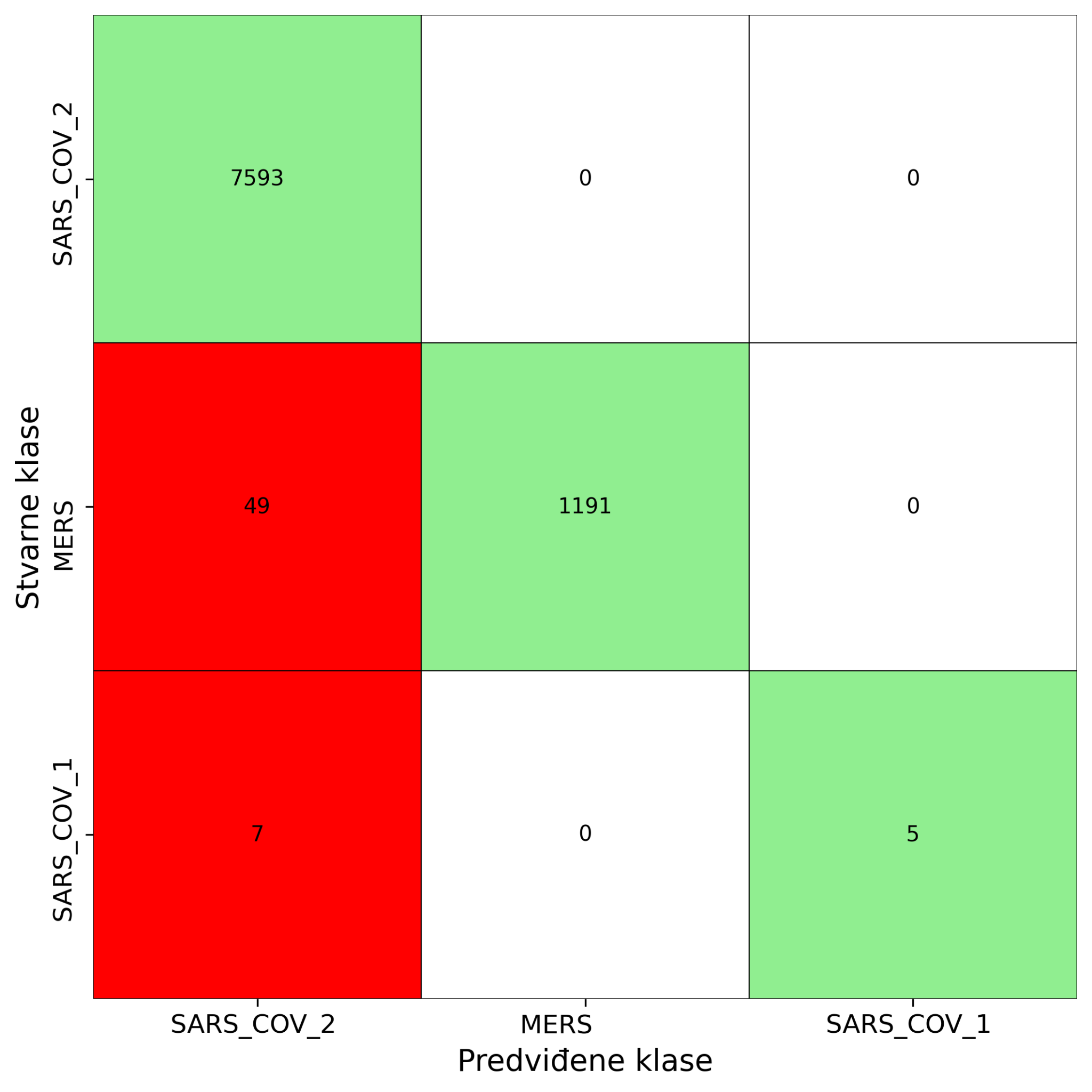
**Ocena modela:**Model je evaluiran na trening i test skupu, pri čemu su dobijene sledeće vrednosti tačnosti:

* **Trening skup:** 0.993
* **Test skup:** 0.989

U nastavku su prikazane matrice konfuzije Modela 2 na trening i test skupu. Ove matrice su sačuvane na sledećim putanjama: *results/confusion\_matrix/train\_hamming\_full\_virus.csv, results/confusion\_matrix/test\_hamming\_full\_virus.csv.*

**

**Tabela 6** Matrica konfuzije Modela 2 na test skupu

**

**Tabela 7** Matrica konfuzije Modela 2 na trening skupu

### 4.1.3 Model 3

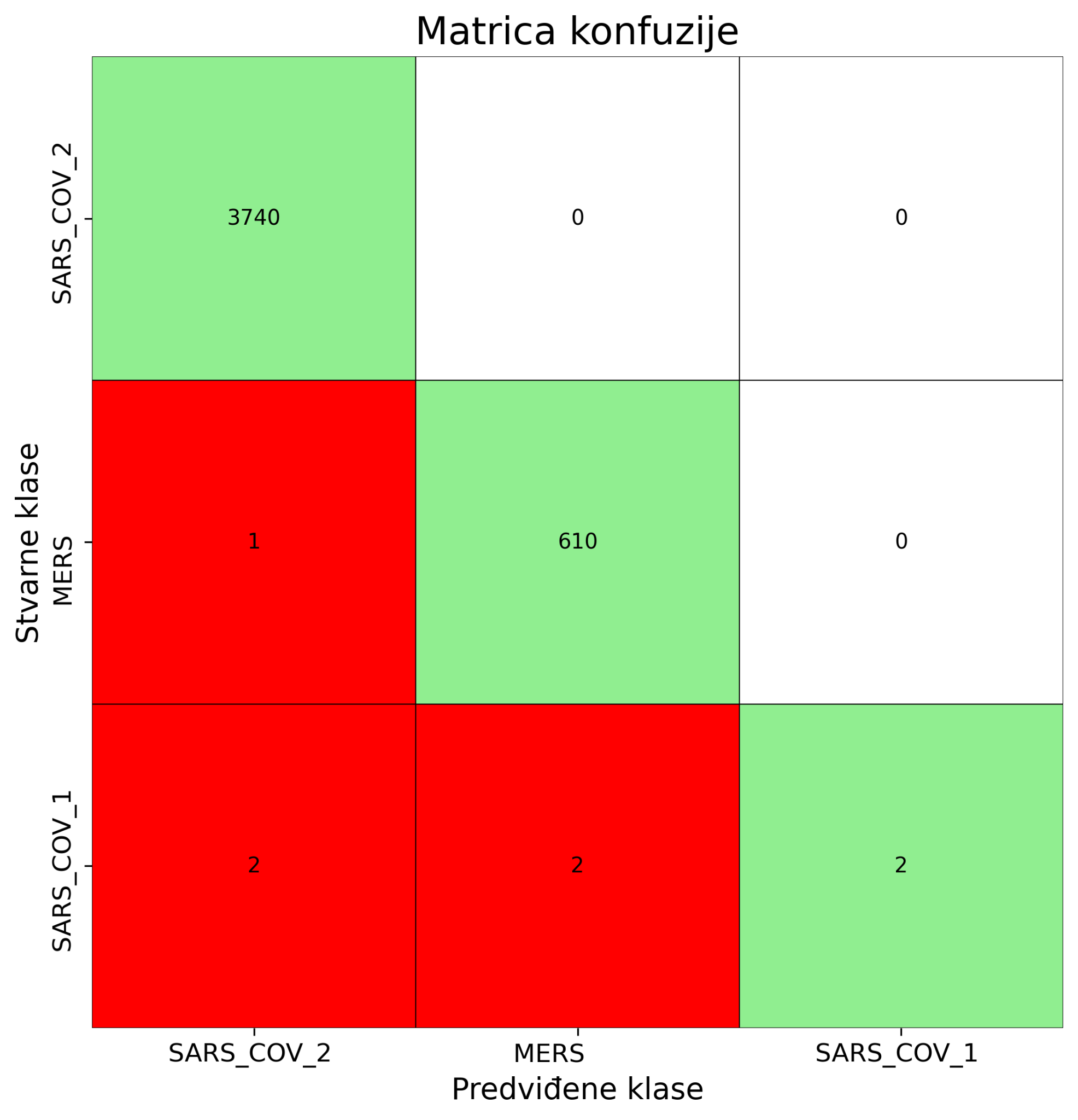
Ovaj model koristi algoritam K-najbližih suseda (*KNeighborsClassifier*) iz biblioteke *scikit-learn* za klasifikaciju tipova virusa. Za izračunavanje rastojanja između sekvenci primenjuje se **P-adično rastojanje**, uz **fiksno odsecanje** sekvenci na dužinu najkraće sekvence u skupu (78 kodona). Broj suseda u modelu postavljen je na **3**.

Implementacija modela nalazi se u Jupyter svesci *source/KNN.ipynb* pod naslovom **Model 3**.

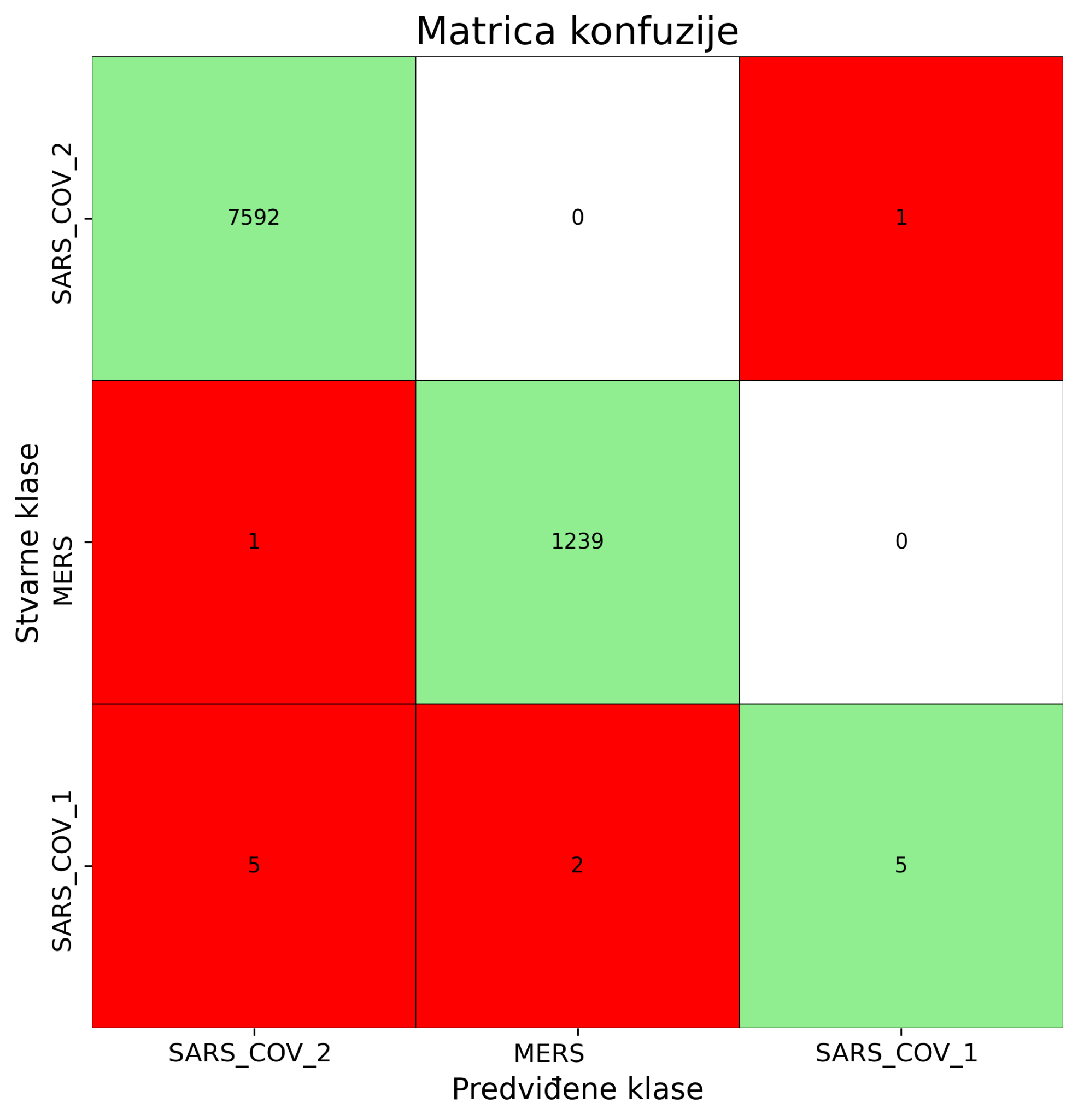
**Ocena modela:**Model je evaluiran na trening i test skupu, pri čemu su dobijene sledeće vrednosti tačnosti:

* **Trening skup:** 0.999
* **Test skup:** 0.999

U nastavku su prikazane matrice konfuzije Modela 3 na trening i test skupu. Ove matrice su sačuvane u CSV datotekama na putanjama *results/confusion\_matrix/train\_padic\_clipped \_virus.csv* i *results/confusion\_matrix/test\_padic\_clipped\_virus.csv.*

**

**Tabela 8** Matrica konfuzije Modela 3 na test skupu

**

**Tabela 9** Matrica konfuzije Modela 3 na trening skupu

### 4.1.4 Model 4

Ovaj model koristi algoritam K-najbližih suseda (*KNeighborsClassifier*) iz biblioteke *scikit-learn* za klasifikaciju tipova virusa. Za izračunavanje rastojanja između sekvenci primenjuje se **Hamingovo rastojanje**, uz **fiksno odsecanje** sekvenci na dužinu najkraće sekvence u skupu (78 kodona). Broj suseda u modelu postavljen je na **3**.

Implementacija modela nalazi se u Jupyter svesci *source/KNN.ipynb* pod naslovom **Model 4**.

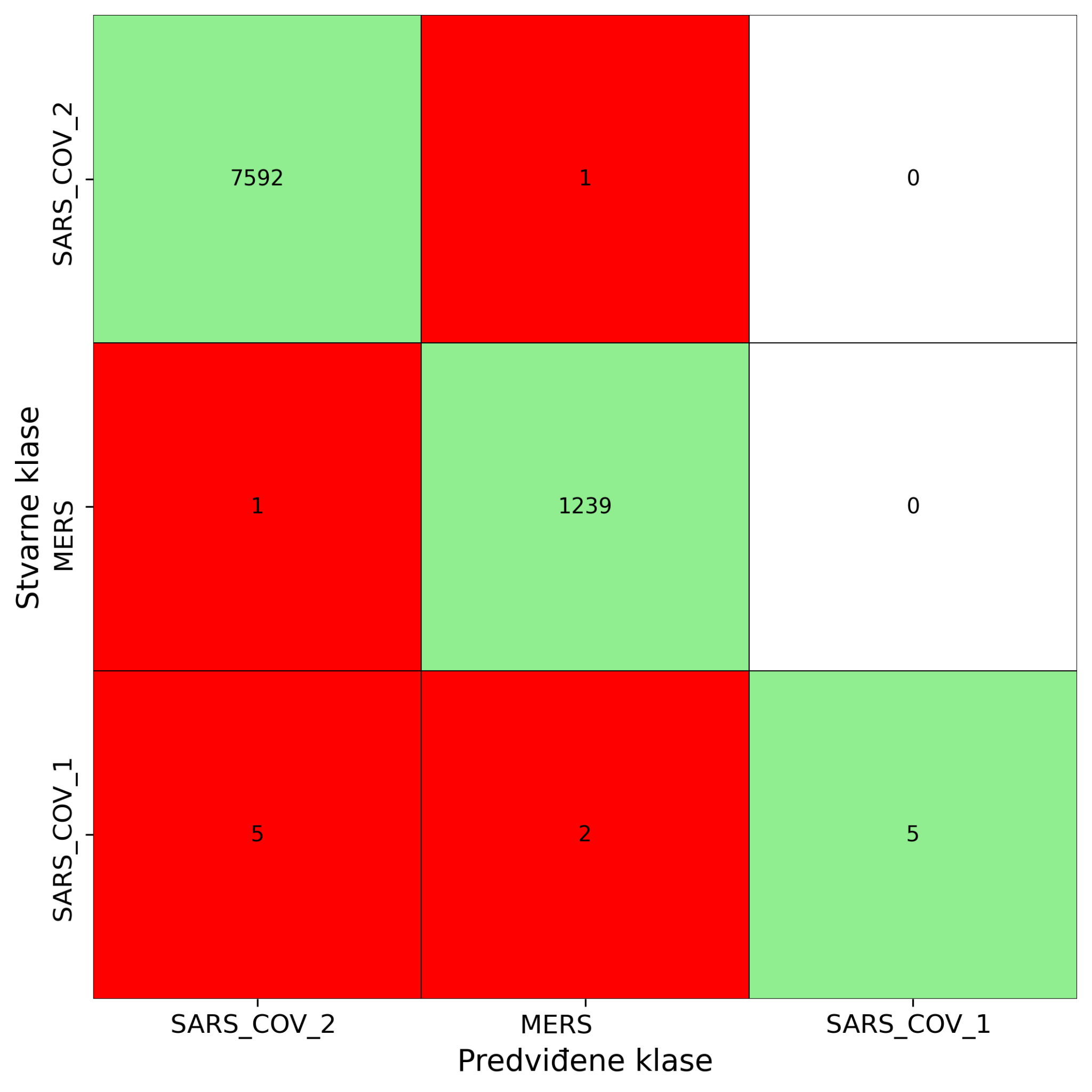
**Ocena modela:**Model je evaluiran na trening i test skupu, pri čemu su dobijene sledeće vrednosti tačnosti:

* **Trening skup:** 0.999
* **Test skup:** 0.999

U nastavku su prikazane matrice konfuzije Modela 3 na trening i test skupu. Ove matrice su sačuvane u CSV datotekama na putanjama *results/confusion\_matrix/train\_hamming\_clipped \_virus.csv* i *results/confusion\_matrix/test\_hamming\_clipped\_virus.csv.*



**Tabela 10**  Matrica konfuzije Modela 4 na test skupu



**Tabela 11**  Matrica konfuzije Modela 4 na trening skupu

## 4.2 Modeli klasifikacije po tipu proteina

U narednim podpoglavljima biće prikazani modeli klasifikacije po tipu proteina, koristeći različlite algoritme i rastojanja između sekvenci.

### 4.2.1 Model 5

Ovaj model primenjuje algoritam K-najbližih suseda (*KNeighborsClassifier*) iz biblioteke *scikit-learn* za klasifikaciju tipova proteina. Model koristi **P-adičnu metriku** za izračunavanje rastojanja između sekvenci, uz primenu **adaptivnog odsecanja**. Parametar broja suseda postavljen je na **3**.

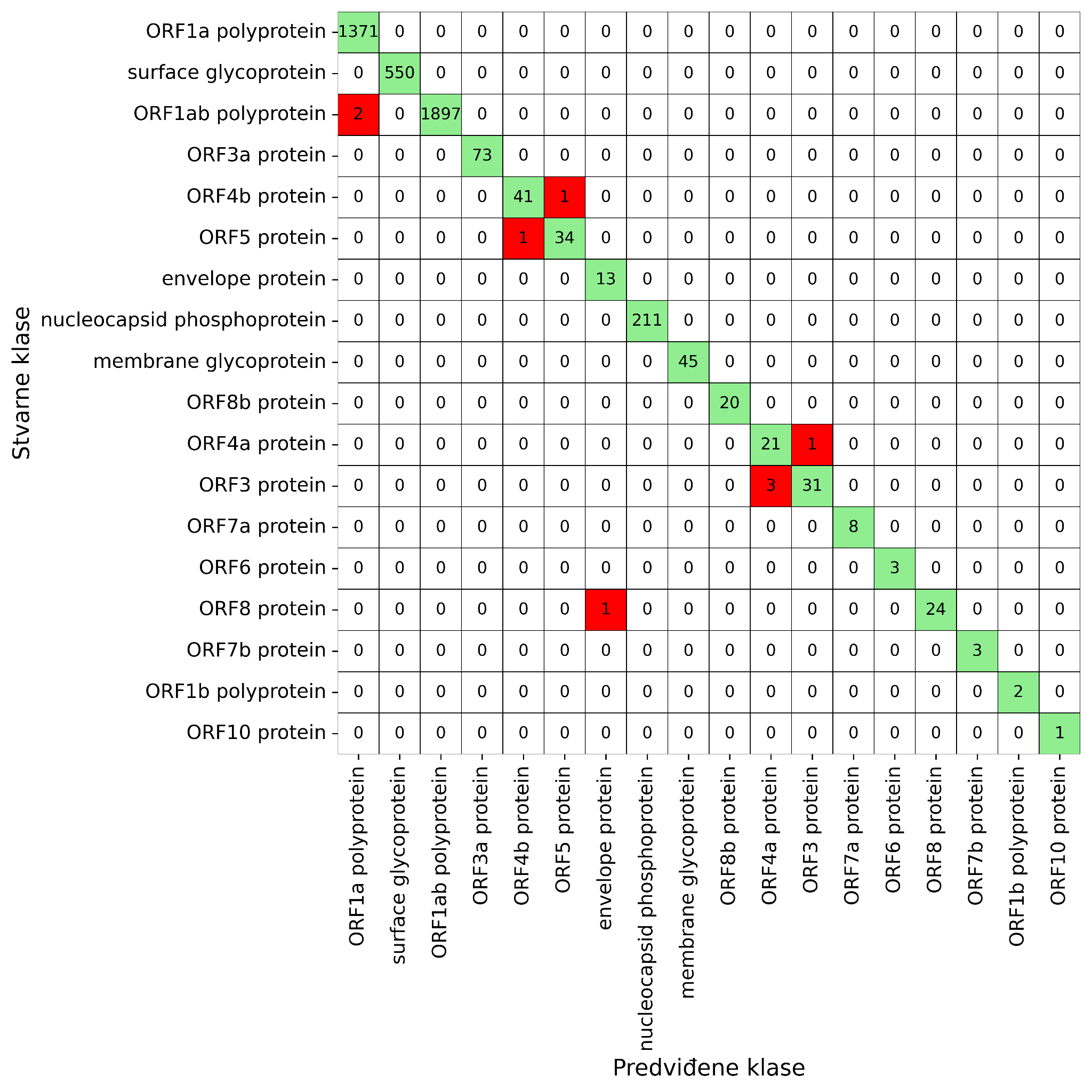
Implementacija modela nalazi se u Jupyter svesci *source/KNN.ipynb* pod naslovom **Model 5**.

**Ocena modela:**Dobijene su sledeće vrednosti tačnosti modela na trening i test skupu:

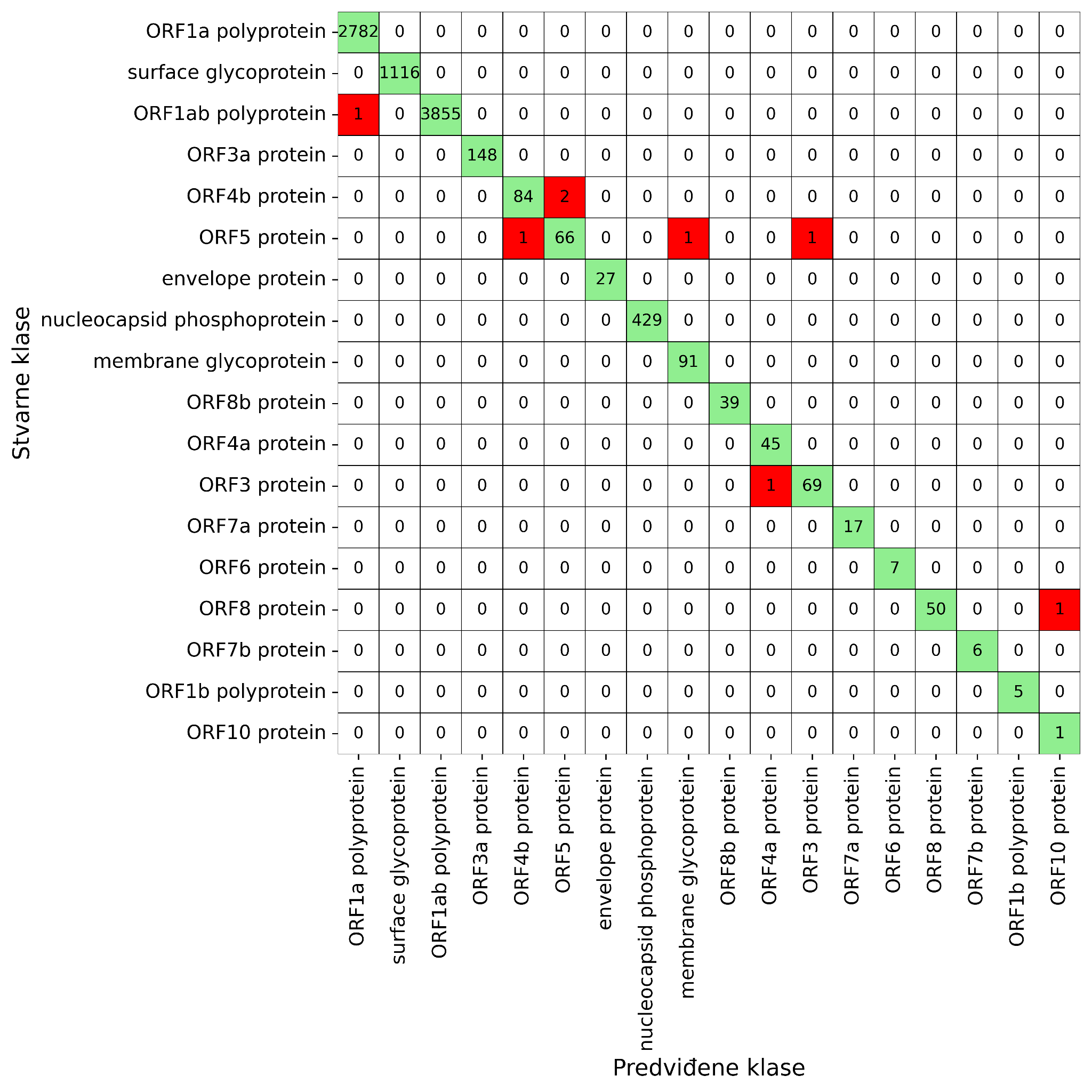
* **Trening skup:** 0.999
* **Test skup:** 0.997

U nastavku su prikazane matrice konfuzije Modela 5 na trening i test skupu. Takođe, ove matrice su sačuvane na putanjama:

* *results/confusion\_matrix/train\_padic\_full\_protein.csv*
* *results/confusion\_matrix/test\_padic\_full\_protein.csv*

****

**Tabela 12** Matrica konfuzije Modela 5 na test skupu



**Tabela 13** Matrica konfuzije Modela 5 na trening skupu

### 4.2.2 Model 6

Ovaj model koristi algoritam K-najbližih suseda (*KNeighborsClassifier*) iz biblioteke *scikit-learn* za klasifikaciju tipova proteina. Za izračunavanje rastojanja između sekvenci primenjuje se **Hamingovo rastojanje**, uz **adaptivno odsecanje** sekvenci. Broj suseda u modelu postavljen je na **3**.

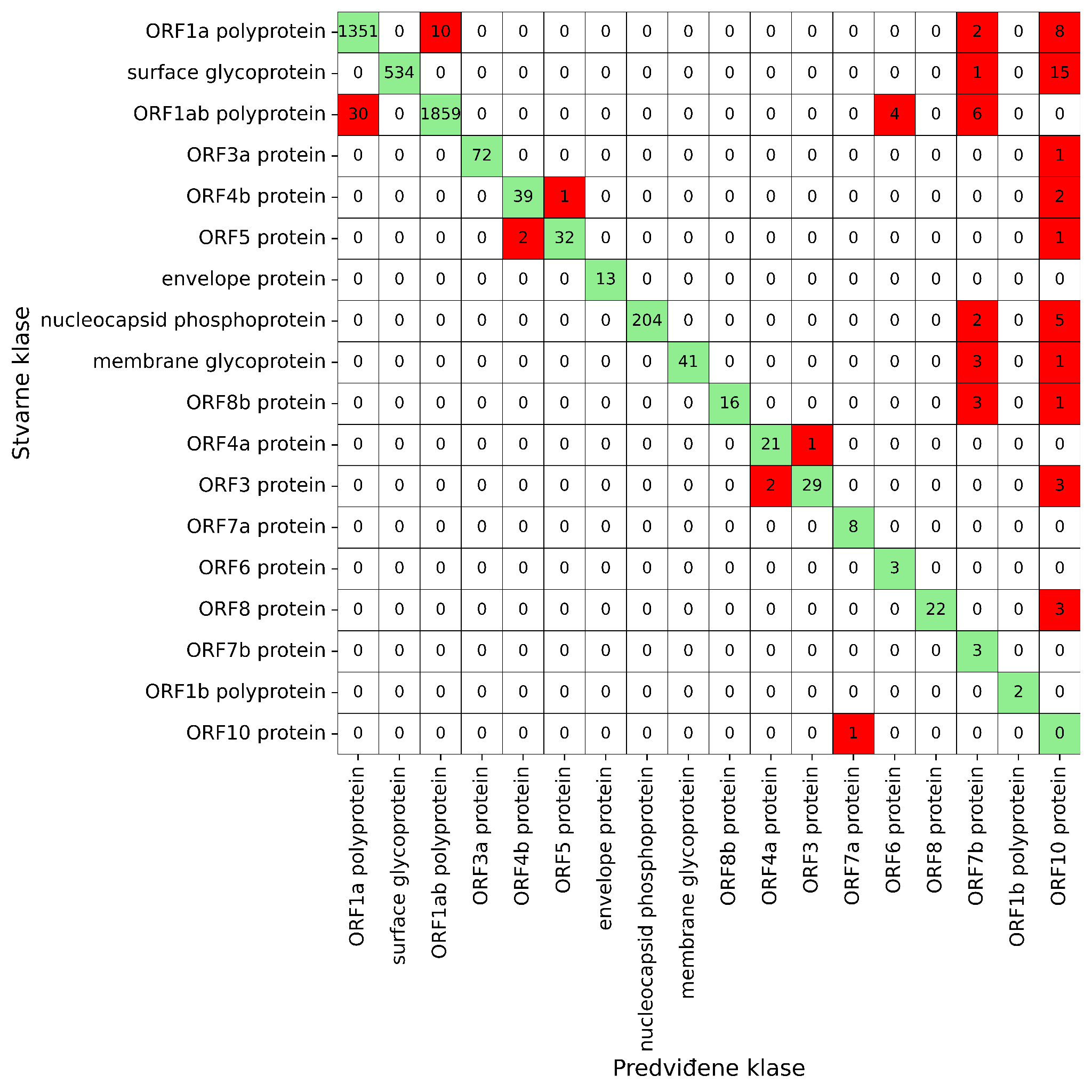
Implementacija modela nalazi se u Jupyter svesci *source/KNN.ipynb* pod naslovom **Model 6**.

**Ocena modela:**Model je evaluiran na trening i test skupu, pri čemu su dobijene sledeće vrednosti tačnosti:

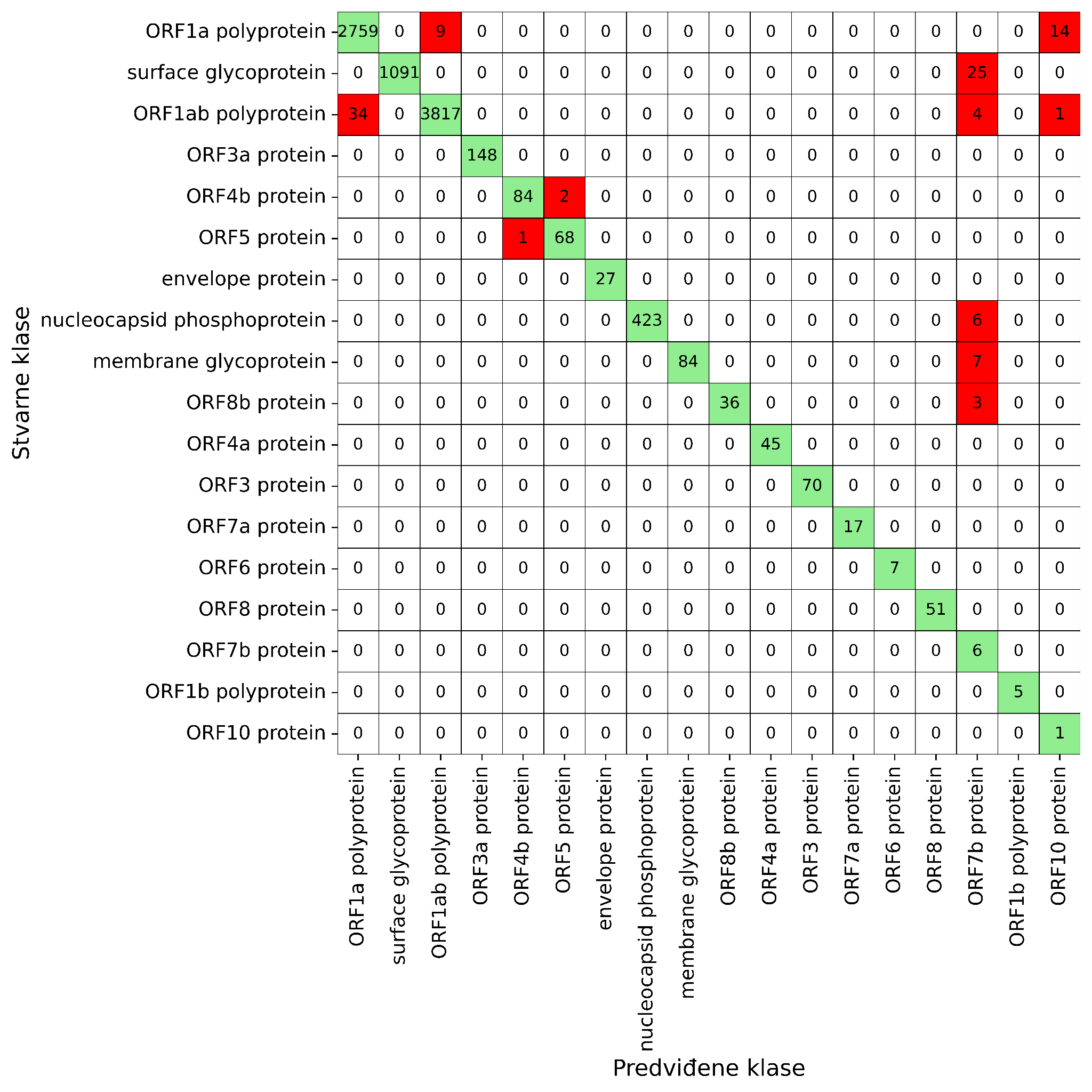
* **Trening skup:** 0.993
* **Test skup:** 0.975

U nastavku su prikazane matrice konfuzije Modela 6 na trening i test skupu. Ove matrice su sačuvane na sledećim putanjama:

* *results/confusion\_matrix/train\_hamming\_full\_protein.csv*
* *results/confusion\_matrix/test\_hamming\_full\_protein.csv*

****

**Tabela 14** Matrica konfuzije Modela 6 na test skupu

****

**Tabela 15** Matrica konfuzije Modela 6 na trening skupu

### 4.2.3 Model 7

Ovaj model koristi algoritam K-najbližih suseda (*KNeighborsClassifier*) iz biblioteke *scikit-learn* za klasifikaciju tipova proteina. Za izračunavanje rastojanja između sekvenci primenjuje se **P-adično rastojanje**, uz **fiksno odsecanje** sekvenci na dužinu najkraće sekvence u skupu (*78 kodona*). Broj suseda u modelu postavljen je na **3**.

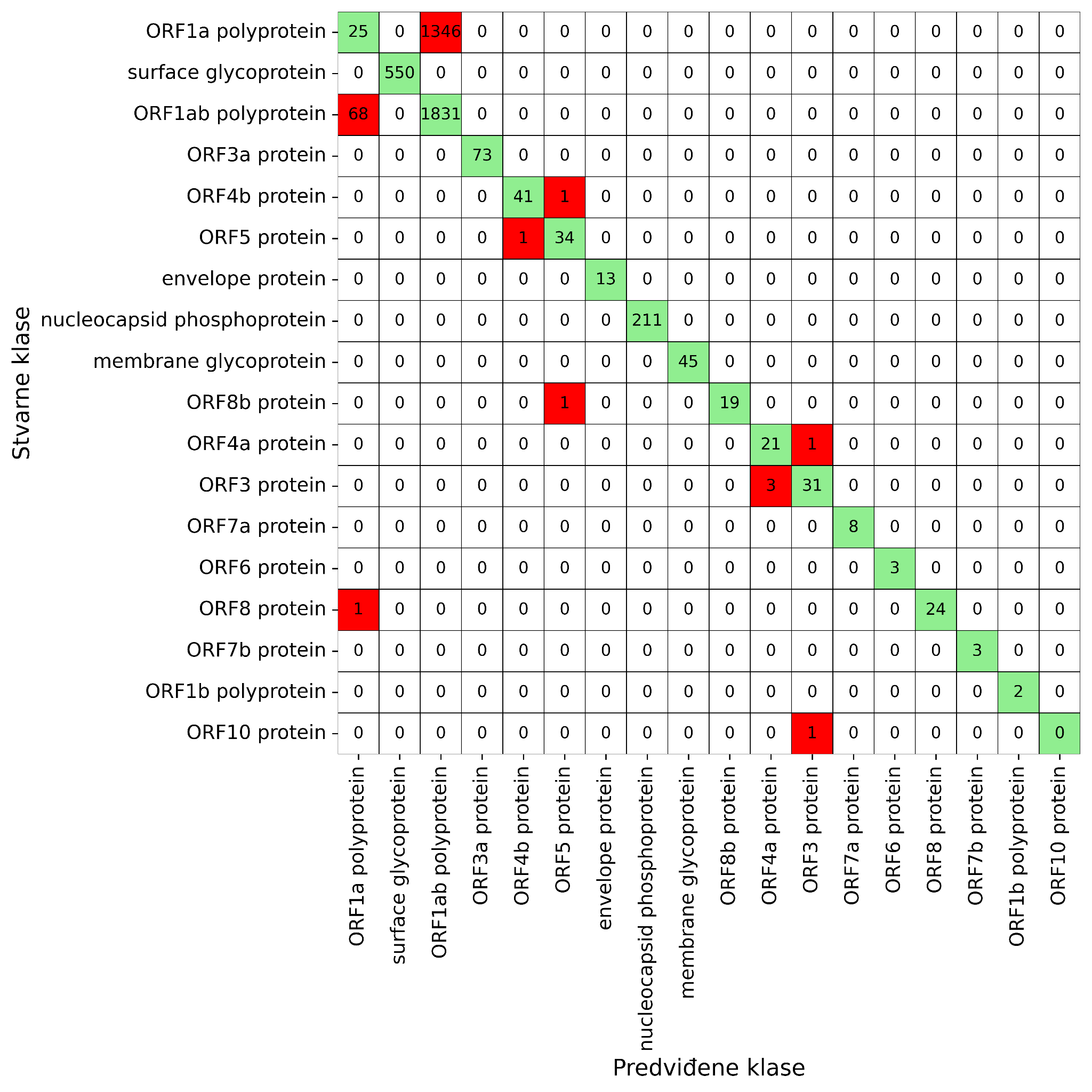
Implementacija modela nalazi se u Jupyter svesci *source/KNN.ipynb* pod naslovom **Model 7**.

**Ocena modela:**Model je evaluiran na trening i test skupu, pri čemu su dobijene sledeće vrednosti tačnosti:

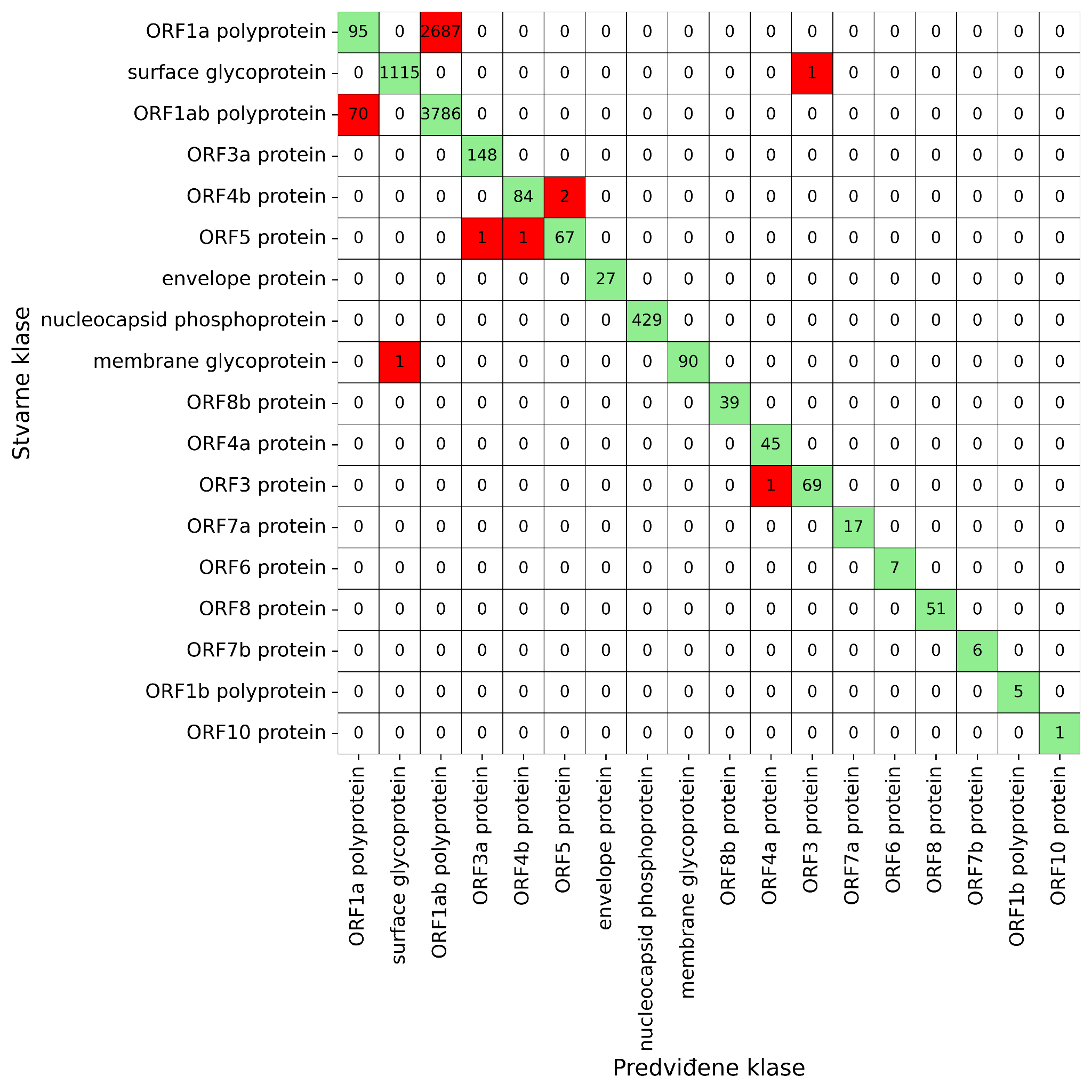
* **Trening skup:** 0.687
* **Test skup:** 0.673

U nastavku su prikazane matrice konfuzije Modela 7 na trening i test skupu. Ove matrice su sačuvane u CSV datotekama na putanjama:

* *results/confusion\_matrix/train\_padic\_clipped\_protein.csv*
* *results/confusion\_matrix/test\_padic\_clipped\_protein.csv*

****

**Tabela 16** Matrica konfuzije Modela 7 na test skupu



**Tabela 17** Matrica konfuzije Modela 7 na trening skupu

### 4.2.4 Model 8

Ovaj model koristi algoritam K-najbližih suseda (*KNeighborsClassifier*) iz biblioteke *scikit-learn* za klasifikaciju tipova proteina. Za izračunavanje rastojanja između sekvenci primenjuje se **Hamingovo rastojanje**, uz **fiksno odsecanje** sekvenci na dužinu najkraće sekvence u skupu (*78 kodona*). Broj suseda u modelu postavljen je na **3**.

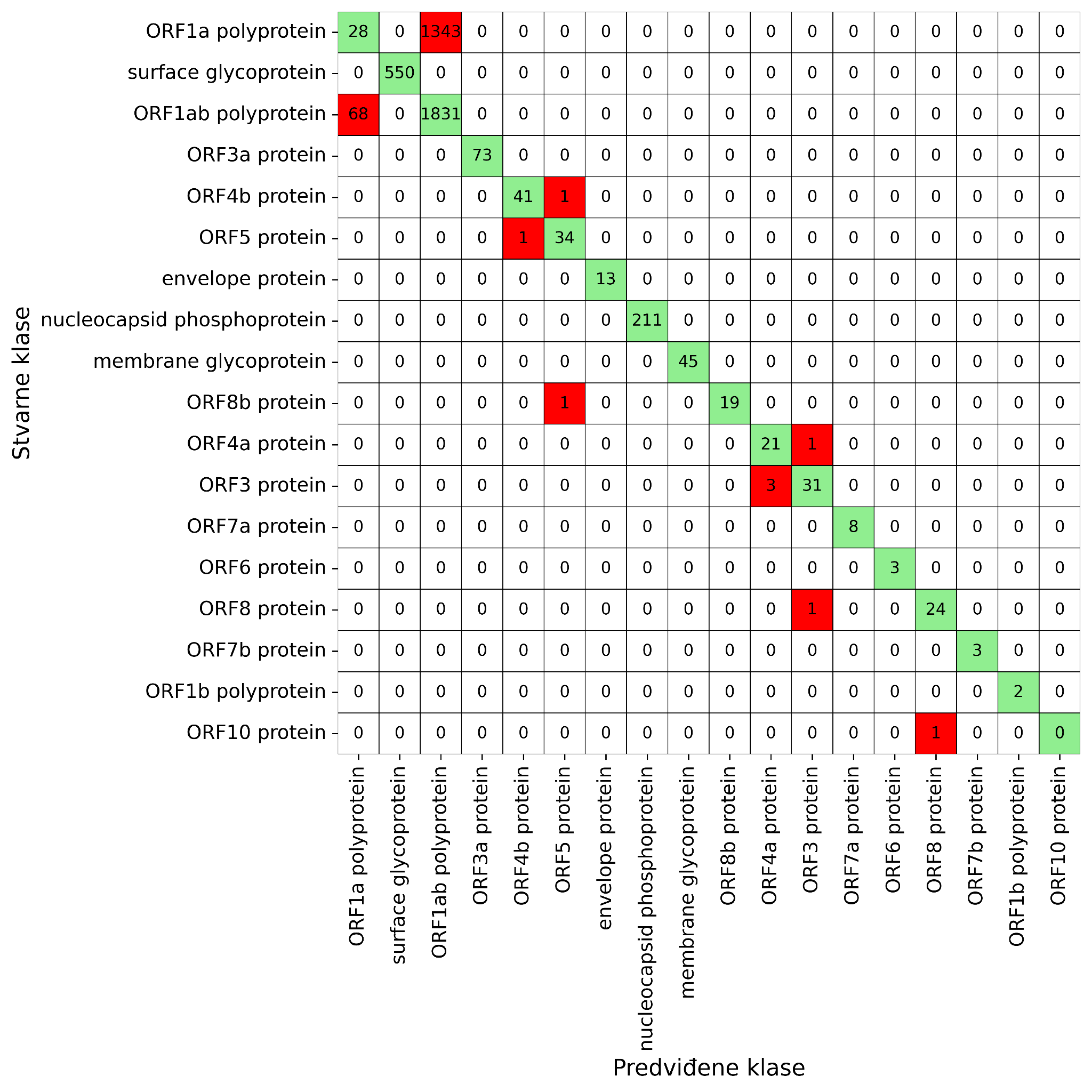
Implementacija modela nalazi se u Jupyter svesci *source/KNN.ipynb* pod naslovom **Model 8**.

**Ocena modela:**Model je evaluiran na trening i test skupu, pri čemu su dobijene sledeće vrednosti tačnosti:

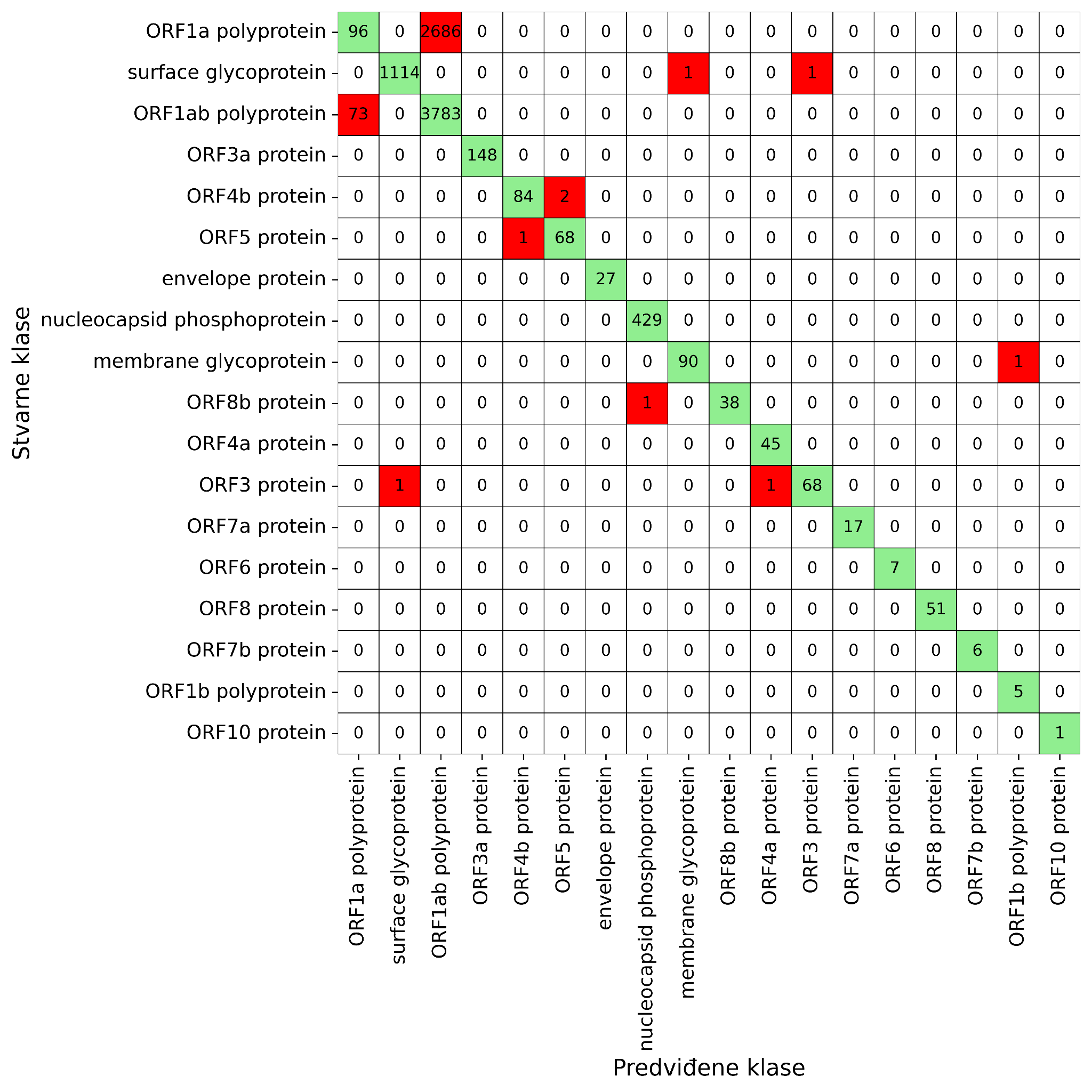
* **Trening skup:** 0.687
* **Test skup:** 0.674

U nastavku su prikazane matrice konfuzije Modela 8 na trening i test skupu. Ove matrice su sačuvane u CSV datotekama na putanjama:

* *results/confusion\_matrix/train\_hamming\_clipped\_protein.csv*
* *results/confusion\_matrix/test\_hamming\_clipped\_protein.csv*

****

**Tabela 18** Matrica konfuzije Modela 8 na test skupu



**Tabela 19** Matrica konfuzije Modela 8 na trening skupu

## 4.3 Modeli klasifikacije po soju SARS2 virusa

U narednim podpoglavljima biće prikazani modeli klasifikacije po soju SARS2 virusa, koristeći različite algoritme i rastojanja između sekvenci.

### 4.3.1 Model 9

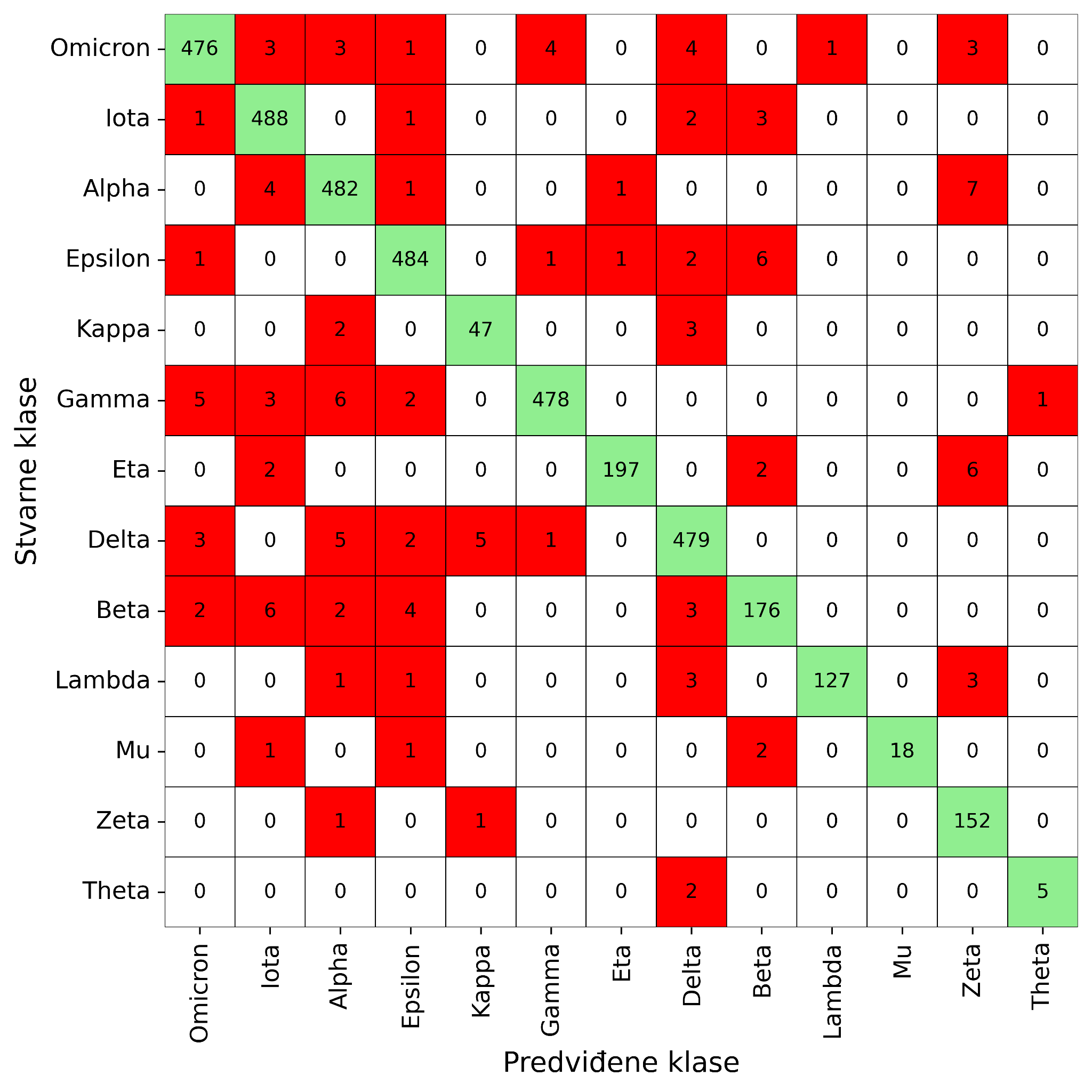
Ovaj model primenjuje algoritam **K-najbližih suseda (KNeighborsClassifier)** iz biblioteke **scikit-learn** za klasifikaciju sojeva SARS2 virusa. Model koristi **P-adičnu metriku** za izračunavanje rastojanja između sekvenci, uz primenu **adaptivnog odsecanja**. Parametar broja suseda postavljen je na **3**.

Implementacija modela nalazi se u Jupyter svesci *source/KNN.ipynb* pod naslovom **Model 9**.

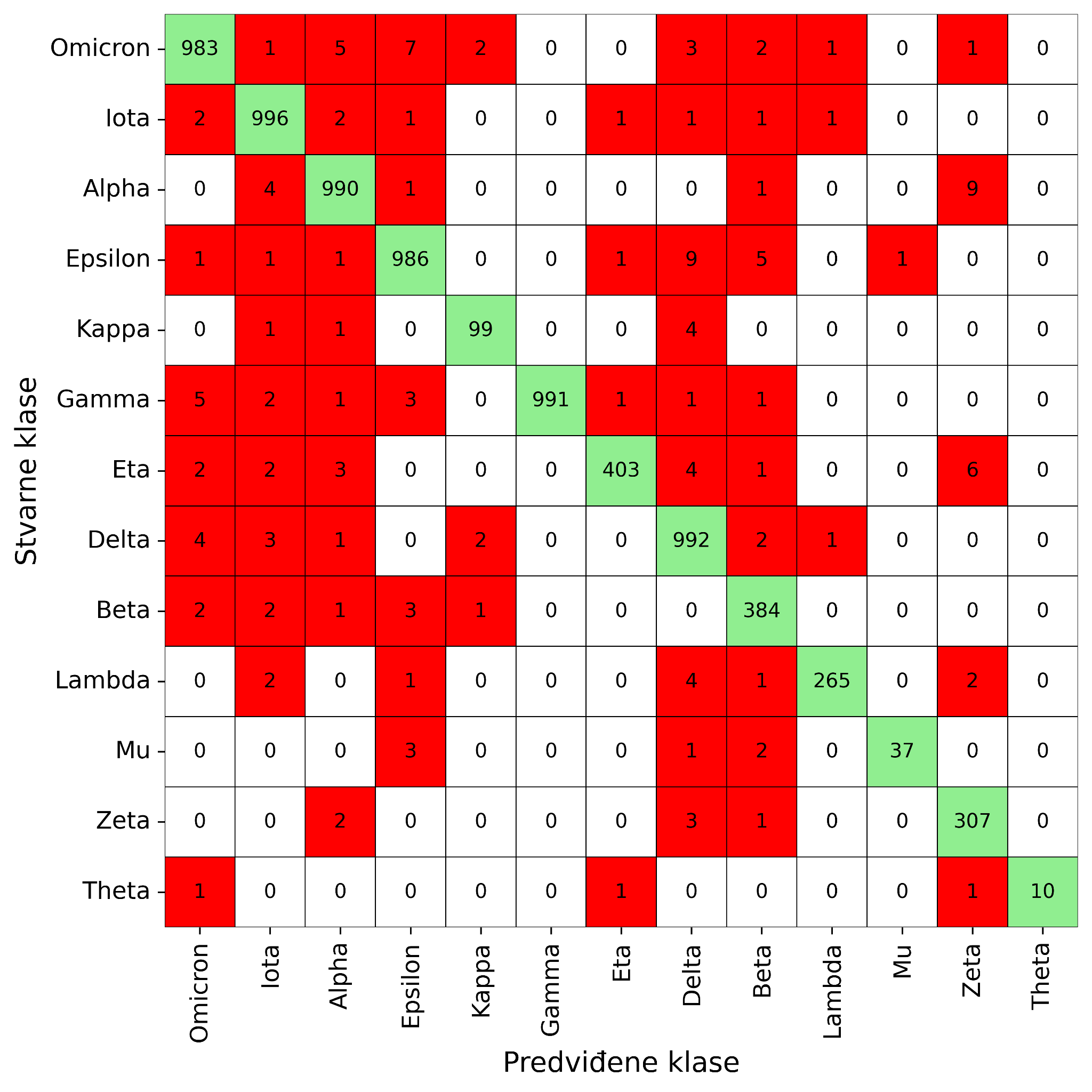
**Ocena modela:**Dobijene su sledeće vrednosti tačnosti modela na trening i test skupu:

* **Test skup:** 0.965
* **Trening skup:** 0.980

U nastavku su prikazane matrice konfuzije Modela 9 na trening i test skupu. Takođe, ove matrice su sačuvane na putanjama: *results/confusion\_matrix/train\_padic\_full\_sars2.csv*i *results/confusion\_matrix/test\_padic\_full\_sars2.csv*

****

**Tabela 20** Matrica konfuzije Modela 9 na test skupu



**Tabela 21** Matrica konfuzije Modela 9 na trening skupu

### 4.3.2 Model 10

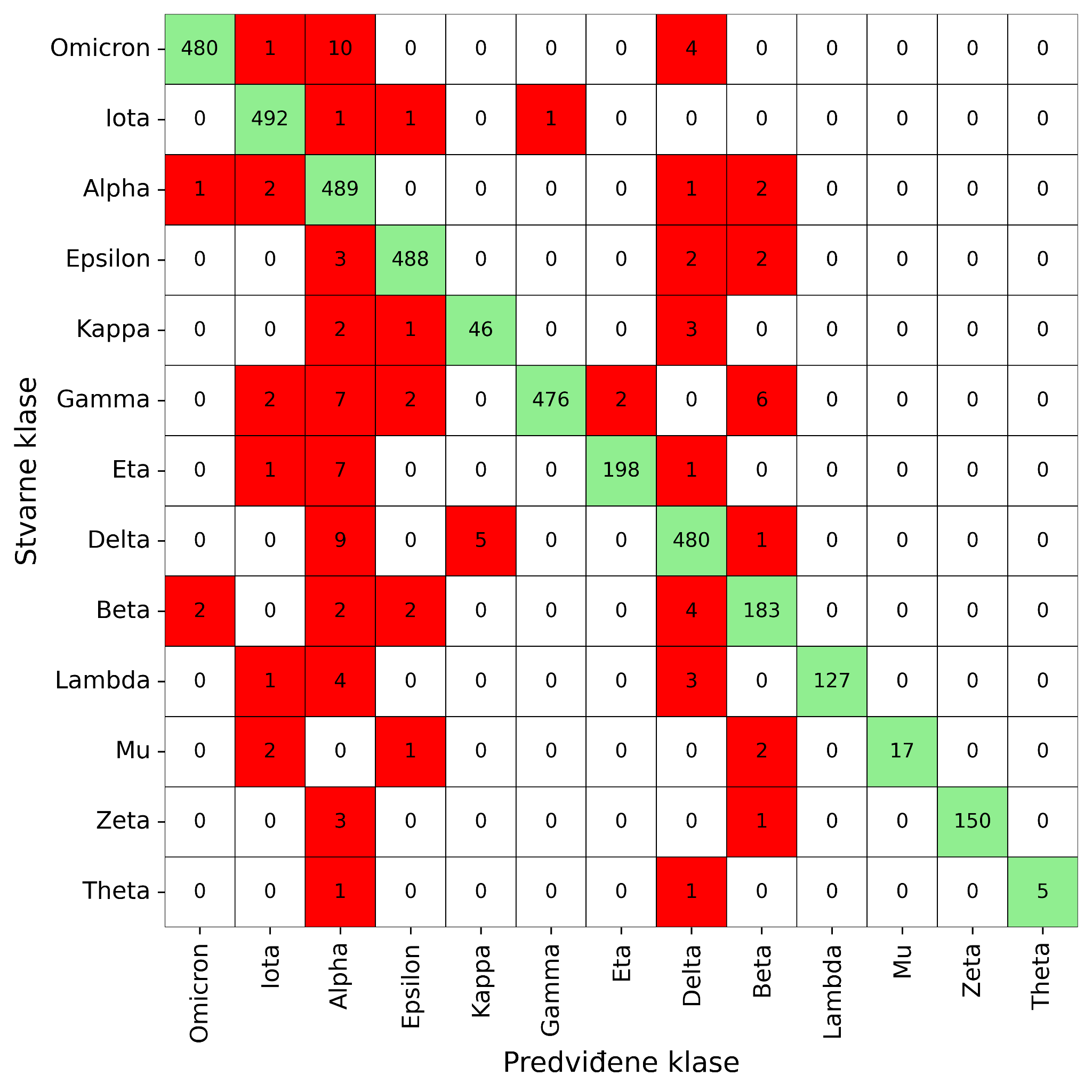
Ovaj model koristi algoritam **K-najbližih suseda (KNeighborsClassifier)** iz biblioteke scikit-learn za klasifikaciju sojeva SARS2 virusa. Za izračunavanje rastojanja između sekvenci primenjuje se **Hamingovo rastojanje**, uz **adaptivno odsecanje sekvenci**. Broj suseda u modelu postavljen je na **3**.

Implementacija modela nalazi se u Jupyter svesci s*ource/KNN.ipynb* pod naslovom Model 10.

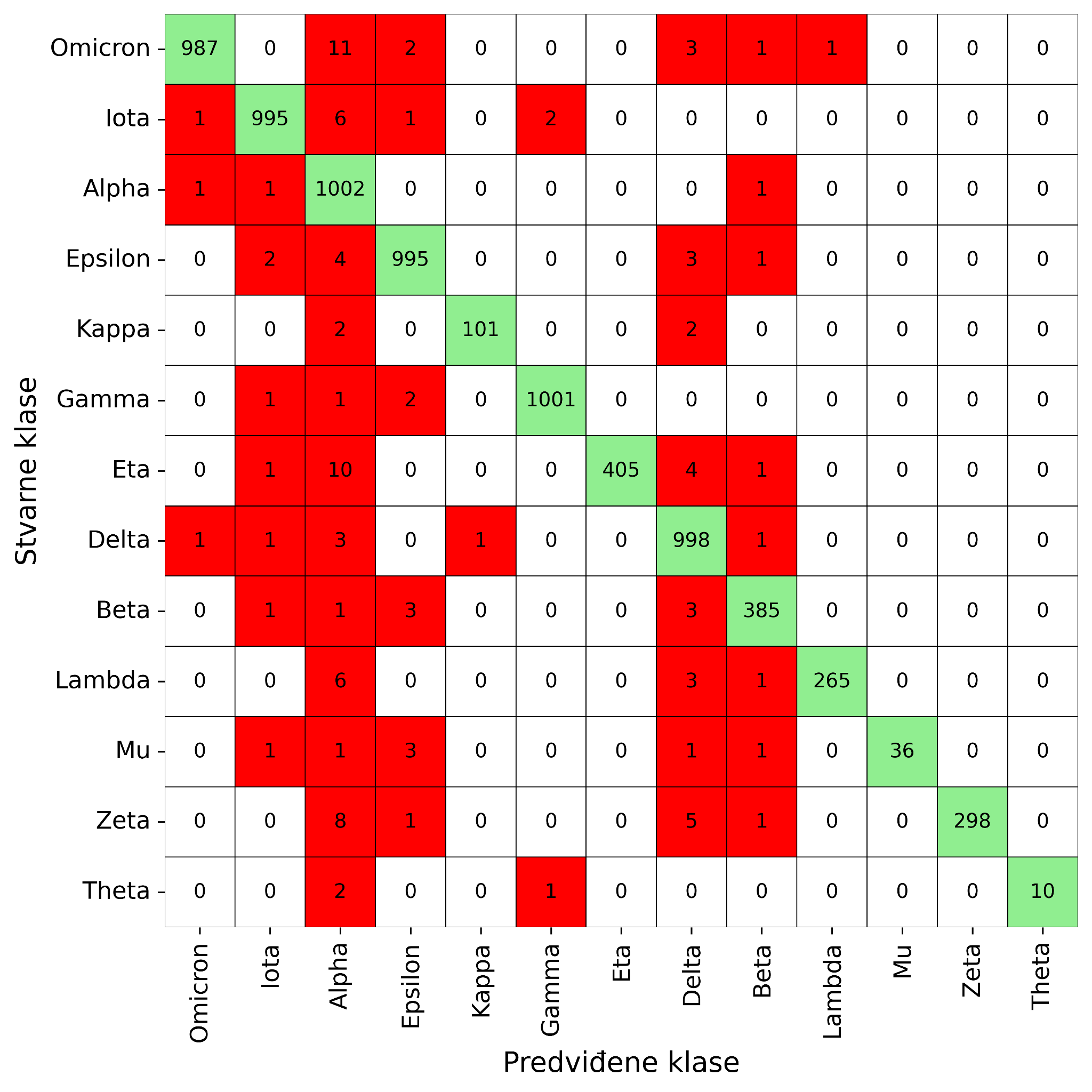
**Ocena modela:**Model je evaluiran na trening i test skupu, pri čemu su dobijene sledeće vrednosti tačnosti:

* **Test skup:** 0.970
* **Trening skup:** 0.985

U nastavku su prikazane matrice konfuzije Modela 10 na trening i test skupu. Ove matrice su sačuvane na sledećim putanjama: *results/confusion\_matrix/train\_hamming\_full\_sars2.csv*i *results/confusion\_matrix/test\_hamming\_full\_sars2.csv*

****

**Tabela 22** Matrica konfuzije Modela 10 na test skupu



**Tabela 23** Matrica konfuzije Modela 10 na trening skupu

### 

### 4.3.3 Model 11

Ovaj model koristi algoritam **K-najbližih suseda (KNeighborsClassifier)** iz biblioteke **scikit-learn** za klasifikaciju sojeva SARS2 virusa. Za izračunavanje rastojanja između sekvenci primenjuje se **P-adično rastojanje**, uz **fiksno odsecanje sekvenci** na dužinu najkraće sekvence u skupu (**78 kodona**). Broj suseda u modelu postavljen je na **3**.

Implementacija modela nalazi se u Jupyter svesci s*ource/KNN.ipynb* pod naslovom Model 11.

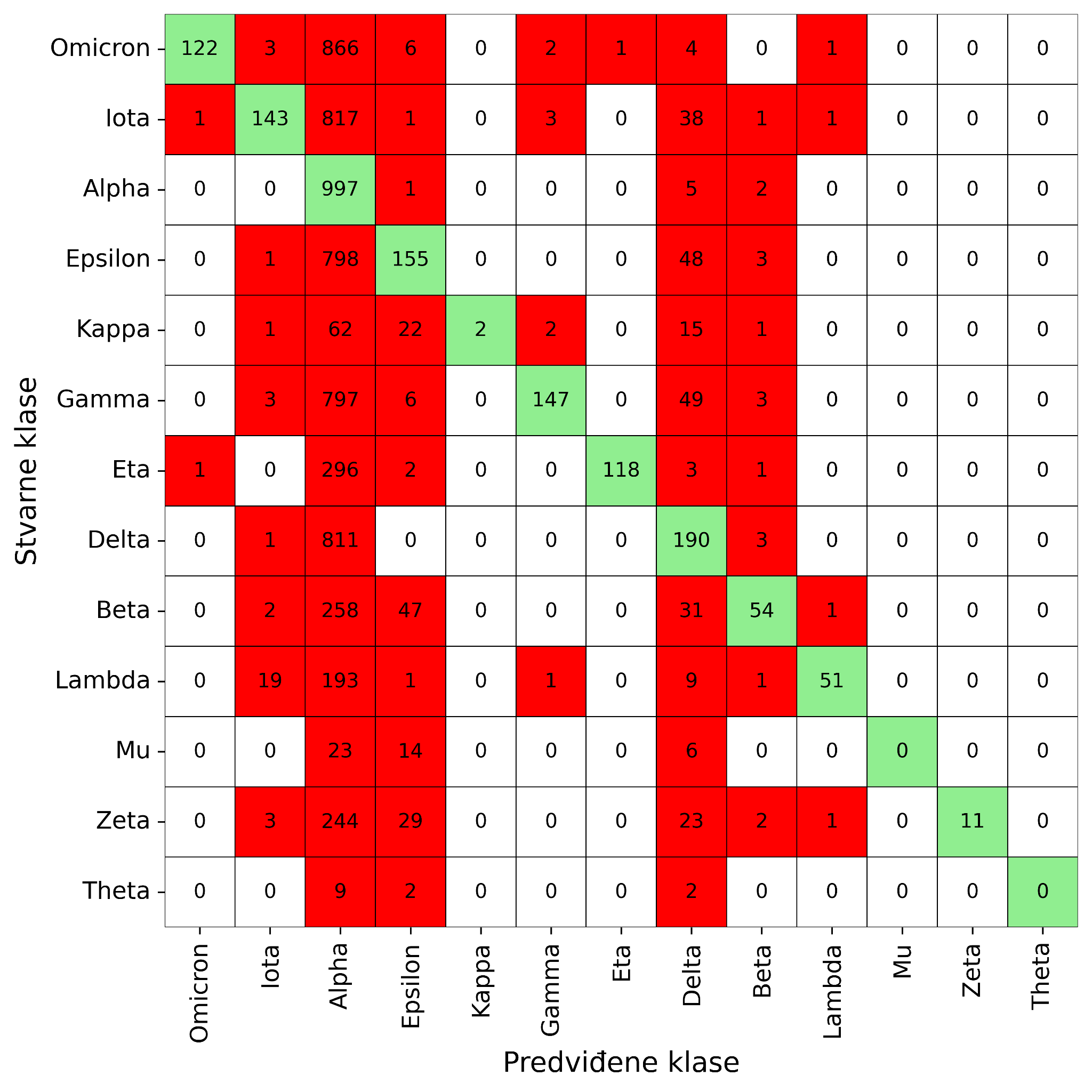
**Ocena modela:**Model je evaluiran na trening i test skupu, pri čemu su dobijene sledeće vrednosti tačnosti:

* **Trening skup:** 0.262
* **Test skup:** 0.255

U nastavku su prikazane **matrice konfuzije Modela 11** na trening i test skupu. Ove matrice su sačuvane u CSV datotekama na putanjama: *results/confusion\_matrix/train\_padic\_ clipped\_sars2.csv* i *results/confusion\_matrix/test\_padic\_clipped\_sars2.csv .*

**

**Tabela 24** Matrica konfuzije Modela 11 na test skupu



**Tabela 25** Matrica konfuzije Modela 11 na trening skupu

### 

### 4.3.4 Model 12

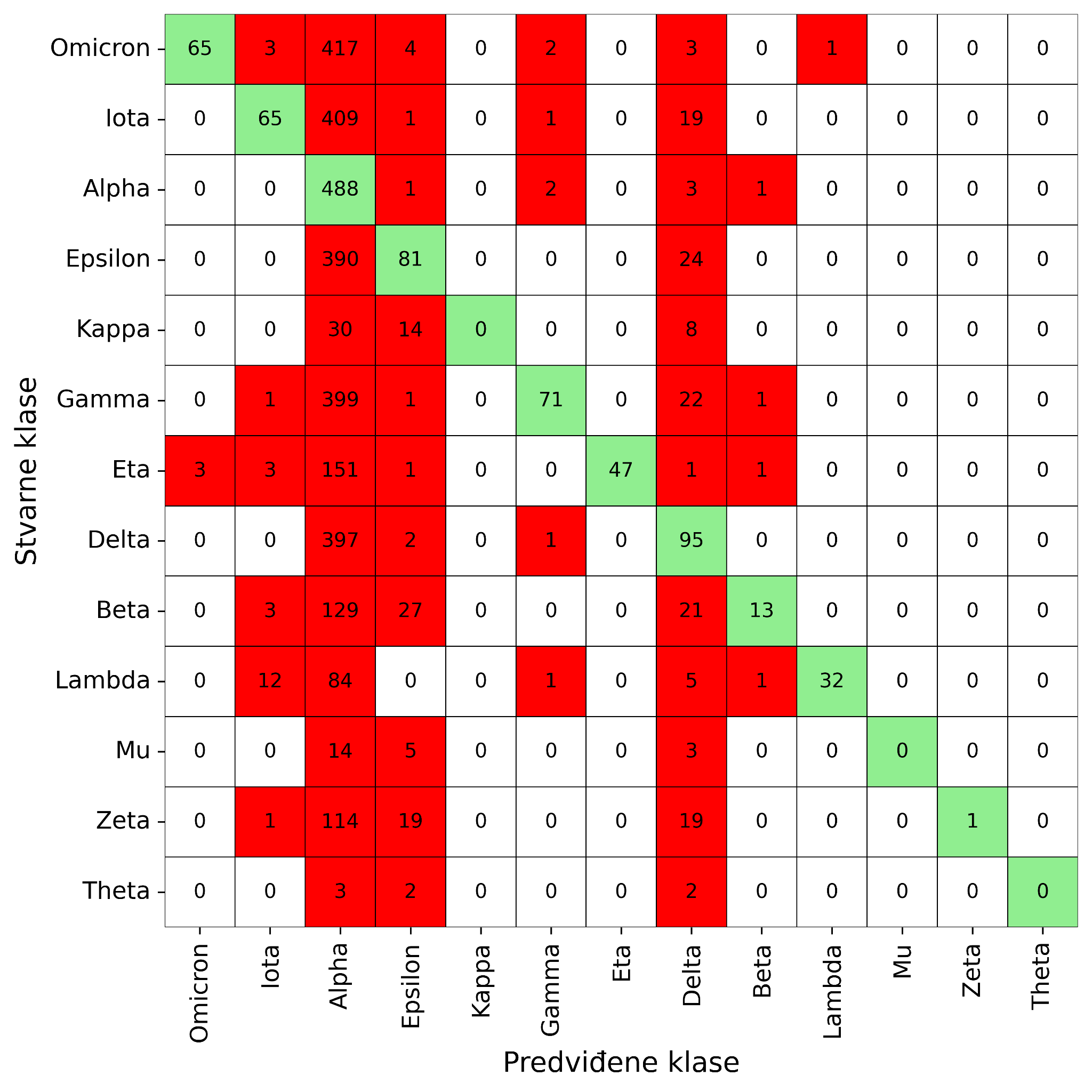
Ovaj model koristi algoritam **K-najbližih suseda (KNeighborsClassifier)** iz biblioteke **scikit-learn** za klasifikaciju sojeva SARS2 virusa. Za izračunavanje rastojanja između sekvenci primenjuje se **Hamingovo rastojanje**, uz **fiksno odsecanje sekvenci** na dužinu najkraće sekvence u skupu (**78 kodona**). Broj suseda u modelu postavljen je na **3**.

Implementacija modela nalazi se u Jupyter svesci *source/KNN.ipynb* pod naslovom Model 12.

**Ocena modela:**Model je evaluiran na trening i test skupu, pri čemu su dobijene sledeće vrednosti tačnosti:

* **Test skup:** 0.256
* **Trening skup:** 0.263

U nastavku su prikazane matrice konfuzije Modela 12 na trening i test skupu. Ove matrice su sačuvane u CSV datotekama na putanjama: *results/confusion\_matrix/train\_hamming\_ clipped\_sars2.csv* i *results/confusion\_matrix/test\_hamming\_clipped\_sars2.csv*



**Tabela 26** Matrica konfuzije Modela 12 na test skupu



**Tabela 27** Matrica konfuzije Modela 12 na trening skupu

## 4.4 Unakrsna provera modela

Izvršena je unakrsna provera hiperparametra k modela za sve 3 klasifikacije. Provereni su modeli za k = 2,...,9. Originalni skup podataka se delio na 4 podskupa (skoro) jednake veličine i svaki od podskupova je korišćen kao validacioni skup. Dobijene su sledeće prosešne preciznosti:

* Za klasifikacija po tipu virusa:  
  + K=2 - 0.9975
  + **K=3 - 0.99775**
  + **K=4 - 0.99775**
  + K=5 - 0.997
  + K=6 - 0.99675
  + K=7 - 0.9965
  + K=8 - 0.9965
  + K=9 - 0.99625

Opaska: Svi klasifikatori na svakom od validacionih skupova su imali preciznost preko 99%.

* Za klasifikacija po tipu proteina:  
  + K=2 - 0.99525
  + **K=3 - 0.99875**
  + K=4 - 0.9985
  + K=5 - 0.9985
  + K=6 - 0.99825
  + K=7 - 0.99825
  + K=8 - 0.99825
  + K=9 - 0.99825

Opaska: Svi klasifikatori na svakom od validacionih skupova su imali preciznost preko 99%.

* Za klasifikaciju sars2 korona virusa po Svetskoj zdravstvenoj organizaciji:  
  + K=2 - 0.94825
  + K=3 - 0.951
  + **K=4 - 0.95125**
  + K=5 - 0.9505
  + K=6 - 0.9495
  + K=7 - 0.94875
  + K=8 - 0.9455
  + K=9 - 0.9445

Opaska: Svi klasifikatori na svakom od validacionih skupova su imali preciznost preko 89%.

Preciznost je približno ista za svako k iz skupa {2, 3, …, 9}, odakle sledi da odabir broja najbližih suseda (iz zadatog skupa) ne utiče bitno na preciznost klasifikacije.

# 

# 5. Zaključak

Uporedjivanjem modela [4.2.2](#_2p2csry) i modela [4.3.1](#_147n2zr) može se primetiti da model 4.2.2 ima malo veću tačnost (0.99 naspram 0.98), ali i još značajnije, bolju matricu konfuzije. Model 4.3.1 ima problema sa prepoznavanjem proteina ORF10 i ORF8b. Kako su oba modela primenjena na celim sekvencama virusa i jedino se razlikuju u primenjenoj metrici (model 4.2.2 koristi p-adičnu metriku), možemo zaključiti da p-adična metrika ima uticaj na razlikovanje proteina SARS1, SARS2 i MERS koronavirusa.

Uporedjivanjem modela iz [4.2.3](#_3o7alnk) i [4.3.2](#_23ckvvd) može se primetiti da model 4.2.3 ima nešto bolje karakteristike od modela 4.3.2 (tačnost od 0.96 naspram 0.81 i značajno bolju matricu konfuzije. Kako je jedina razlika izmedju ovih modela primenjena metrika, možemo zaključiti da postoji uticaj P-adičnosti na razilke genetskog koda SARS1, SARS2 i MERS koronavirusa.

# 6.Reference

[1] Dragovich,, Branko, and Nataša Ž Mišić. “P-Adic Hierarchical Properties of the Genetic Code.” *BioSystems,*, vol. 185, no. 104017, 2019,

[2]<https://www.ncbi.nlm.nih.gov/labs/virus/vssi/#/virus?SeqType_s=Nucleotide&VirusLineage_ss=Severe%20acute%20respiratory%20syndrome%20coronavirus,%20taxid:2901879>

[3] <https://docs.python.org/3/library/pickle.html>

[4]<https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html>

[5] <https://seaborn.pydata.org/generated/seaborn.heatmap.html#>