

Matematički fakultet
Univerzitet u Beogradu

Seminarski rad:
**Uticaj P-adičnosti na razlike genetskog
koda SARS1, SARS2 i MERS
koronavirusa**

Mentor:
Prof. dr Nenad Mitić

Studenti:
Marko Koprivica 61/2021
Milan Bodo 173/2021

1.Uvod	3
2.P-adična norma	3
2.1 P-adično rastojanje	3
2.2 Primena p-adičnih metrika u analizi genetskog koda	4
2.3 P-adično rastojanje izmedju kodona primenjeno u našem istraživanju	5
3.Podaci	5
3.1 Preprocesiranje podataka	7
3.2 Podela podataka	8
4. Modeli	8
4.1 Modeli klasifikacije po tipu virusa	8
4.1.1 Model 1	9
4.1.2 Model 2	11
4.1.3 Model 3	12
4.1.4 Model 4	14
4.2 Modeli klasifikacije po tipu proteina	15
4.2.1 Model 5	15
4.2.2 Model 6	17
4.2.3 Model 7	19
4.2.4 Model 8	21
4.3 Modeli klasifikacije po soju SARS2 virusa	23
4.3.1 Model 9	23
4.3.2 Model 10	25
4.3.3 Model 11	27
4.3.4 Model 12	29
4.4 Optimizacija modela	32
5. Zaključak	34
6.Reference	34

1.Uvod

Genetski kod predstavlja temelj života, čuvajući informacije neophodne za sintezu proteina u svim organizmima. Razumevanje strukture i razlike u genetskim sekvencama virusa od ključnog je značaja za unapređenje dijagnostike, lečenja i prevencije zaraznih bolesti. Ovaj seminarски 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 primjenjuje 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 i metodu potpornih vektora, 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 $\forall m \in \mathbb{Z}$, $m \neq 0$ može zapisati kao $m = p^k \cdot a$ gde je p prost broj, $k \in \{0, 1, 2, \dots\}$ i a ceo broj koji nije delji sa p . P-adična norma broja m u oznaci $|m|_p$ se definiše na sledeći način:

$$|m|_p = p^{-k} \quad [1]$$

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. $|x-y|_p \geq 0$, $|x-y|_p = 0 \Leftrightarrow x=y$
2. $|x-y|_p = |y-x|_p$
3. $|x-y|_p \leq \max \{|x-z|_p, |y-z|_p\}$

Primeri računja p-adičnog rastojanja:

$$|63-3|_2 = |60|_2 = |2^2 \cdot 3 \cdot 5|_2 = 1/4$$

$$|63-3|_3 = |60|_3 = |2^2 \cdot 3 \cdot 5|_3 = 1/3$$

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 $x=x_0x_1x_2$ kodiran u 5-adičnom sistemu može se predstaviti brojem

$$x=x_0+x_1 \cdot 5+x_2 \cdot 5^2.$$

P-adično rastojanje između dva kodona $x=x_0x_1x_2$ i $y=y_0y_1y_2$, za $p=0$ definiše se na sledeći način:

$$d_5(x, y) = \left| (x_0 + x_1 \cdot 5 + x_2 \cdot 5^2) - (y_0 + y_1 \cdot 5 + y_2 \cdot 5^2) \right|_5 = \begin{cases} 1, & \text{ako } x_0 \neq y_0, \\ \frac{1}{5}, & \text{ako } x_0 = y_0, x_1 \neq y_1, \\ \frac{1}{25}, & \text{ako } x_0 = y_0, x_1 = y_1, x_2 \neq y_2. \end{cases}$$

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

Ovaj izraz prikazuje da su kodoni najbliži (tj. $d_5(x,y)=1/25$ je minimalno) ukoliko se razlikuju isključivo na trećoj poziciji, dok se u slučaju razlikovanja na prvoj poziciji dobija maksimalna vrednost rastojanja $d_5(x,y)=1$.

Pored 5-adične metrike, primena 2-adične metrike dodatno određuje sličnost među nukleotidima, naročito u kontekstu razlikovanja purina (A i G) i pirimidina (C i T). Kombinovanjem ove dve metrike, 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 metriči 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:
 $x_0=A=2, x_1=C=1, x_2=A=2$
- Za kodon y:
 $y_0=A=2, y_1=A=2, y_2=A=2.$

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

$$d_5(x, y) = \left| (2 + 1 \cdot 5 + 2 \cdot 5^2) - (2 + 2 \cdot 5 + 2 \cdot 5^2) \right|_5 = |57 - 62|_5 = |-5|_5 = 1/5$$

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 $5^0=1$, 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 $2^1 \cdot 5^2=1/2 \cdot 1/25$ ili ako ta razlika nije 2 rastojanje se uvećava za $2^0 \cdot 5^2=1 \cdot 1/25$. 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 sekvene proteina odabrane tako da budu nukleotidno kompletne i bez više znacnih 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 Rasподела 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 Preprocesiranje 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 sekvene** – sve sekvene se skraćuju na dužinu najkraće sekvene u skupu podataka (u ovom slučaju 78 kodona).
2. **Adaptivno odsecanje u zavisnosti od poređenih sekvenci** – pri svakom računanju rastojanja između dve sekvene, sekvene se skraćuju na dužinu kraće od njih. Odsečeni deo veće sekvene se ne odbacuje već se distanca povećava za broj kodona (tripleta) u odsečenom delu.

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 Hamingovog 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

KNN

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.

SVM

U svrhu upoređivanja efikasnosti algoritma K-najbližih suseda, dodati su i modeli zasnovani na metodu potpornih vektora sa proizvoljno definisanim jezgrom. Jezgro je zasnovano na p-adičnoj distanci sa [adaptivnim odsecanjem](#), s tim što je jedina razlika znak rezultata (u ovom slučaju dodajemo predznak minus na rezultat). To je iz razloga prirode funkcionisanja jezgra, što je izračunata vrednost veća, to je sličnost 2 instance veća. Jezgro je definisano

preko Gram matrice u funkciji `get_gram(X1, X2, kernel)`, gde parametri X1 i X2 identifikuju sekvence u originalnom skupu podataka, a kernel predstavlja jezgro koje se izračunava za instance X1 i X2. Kernel se izračunava u funkciji `fiveadic_distance_from_matrix(x, y)`, na već opisan način. Funkcija `visualize_confusion_matrix(cm, labels, save_path="")` prima matricu konfuzije (cm) i imena klase (labels) i prikazuje i čuva matricu konfuzije na putanji `results/heatmaps/svm`.

4.1 Modeli klasifikacije po tipu virusa

U narednim podpoglavlјima biće prikazani modeli klasifikacije po tipu virusa, koristeći različite 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 biblioteke `seaborn[5]`. Ovakav način prikaza matrice konfuzije će koristiti u prikazima matrica konfuzije u celom radu.

		SARS_COV_2	MERS	SARS_COV_1
Stvarne klase	SARS_COV_2	3740	0	0
	MERS	3	608	0
SARS_COV_1	SARS_COV_2	3	1	2
	MERS			

Tabela 4 Matrica konfuzije Modela 1 na test skupu

		SARS_COV_2	0	0
Stvarne klase	SARS_COV_2	7593	0	0
	MERS	0	1240	0
SARS_COV_1	SARS_COV_2	5	2	5
	MERS			

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*.

		SARS_COV_2	MERS	SARS_COV_1
Stvarne klase	SARS_COV_2	3740	0	0
	MERS	45	566	0
SARS_COV_1	4	0	2	
Predviđene klase	SARS_COV_2			

Tabela 6 Matrica konfuzije Modela 2 na test skupu

		SARS_COV_2	MERS	SARS_COV_1
Stvarne klase	SARS_COV_2	7593	0	0
	MERS	49	1191	0
SARS_COV_1	SARS_COV_2	7	0	5
	Predviđene klase			

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 sekvene 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*.

Matrica konfuzije			
Stvarne klase	Predviđene klase		
	SARS_COV_2	MERS	SARS_COV_1
SARS_COV_2	3740	0	0
MERS	1	610	0
SARS_COV_1	2	2	2

Tabela 8 Matrica konfuzije Modela 3 na test skupu

Matrica konfuzije			
Stvarne klase	Predviđene klase		
	SARS_COV_2	MERS	SARS_COV_1
SARS_COV_2	7592	0	1
MERS	1	1239	0
SARS_COV_1	5	2	5

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 4 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*.

		SARS_COV_2	MERS	SARS_COV_1
Stvarne klase	SARS_COV_2	3740	0	0
	MERS	1	610	0
SARS_COV_1	SARS_COV_2	3	1	2
	Predviđene klase			

Tabela 10 Matrica konfuzije Modela 4 na test skupu

		SARS_COV_2	MERS	SARS_COV_1
		Stvarne klase		
SARS_COV_2	SARS_COV_2	7592	1	0
	MERS	1	1239	0
SARS_COV_1	SARS_COV_1	5	2	5
	Predviđene klase			

Tabela 11 Matrica konfuzije Modela 4 na trening skupu

4.1.5 Model zasnovan na metodu potpornih vektora

Ovaj model primjenjuje metod potpornih vektora (SVM – Support Vector Machine) iz biblioteke scikit-learn[9] za klasifikaciju tipova virusa.

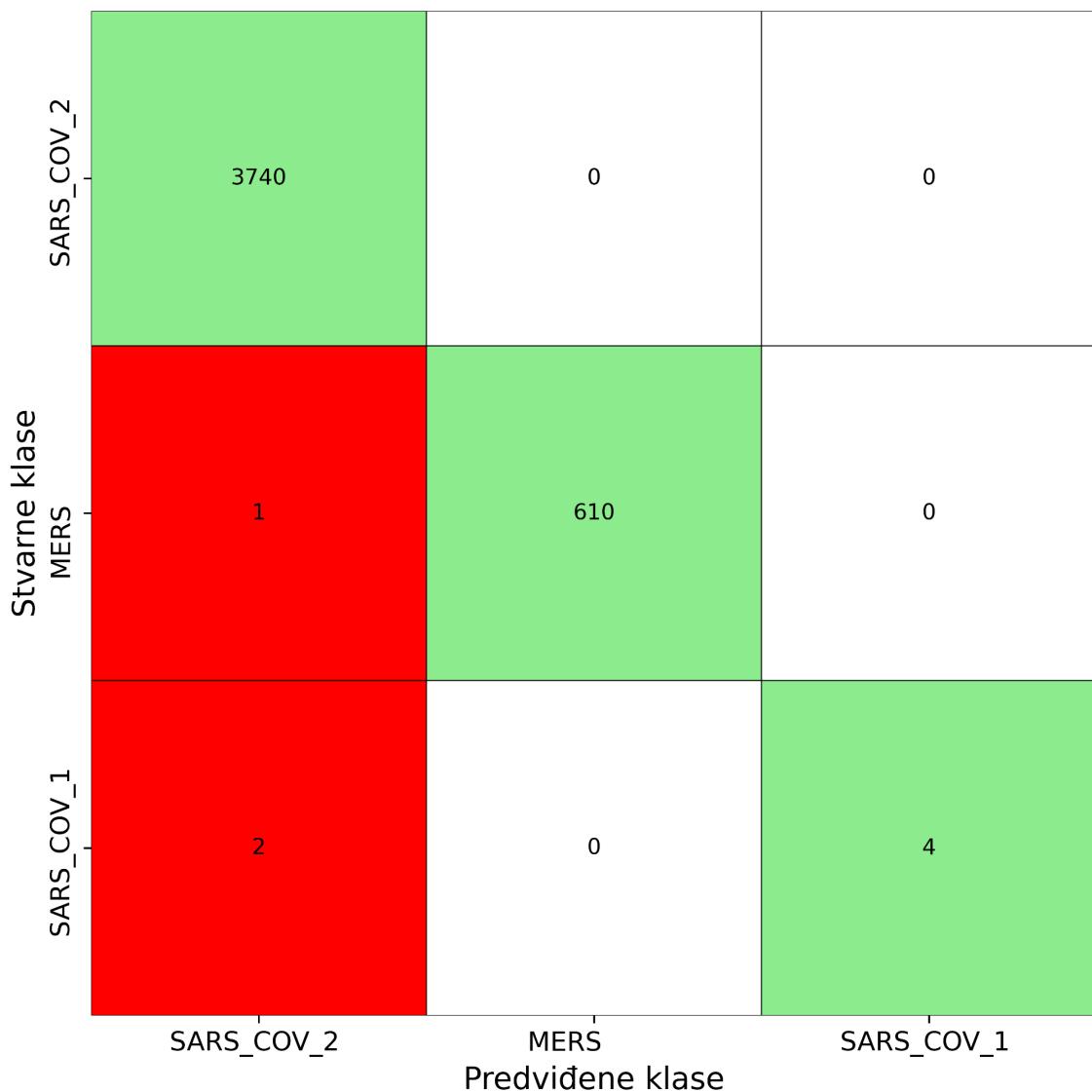
Implementacija modela nalazi se u Jupyter svesci source/svm.ipynb pod naslovom **Model 1**.

Ocena modela:

Dobijene su sledeće vrednosti tačnosti modela na test skupu:

- **Test skup:** 0.999

U nastavku je prikazana matrica konfuzije Modela 1 na test skupu. Ova matrica je sačuvana na putanji *results/heatmaps/svm/virus*.



Možemo primetiti da model zasnovan na metodu potpornih vektora znatno bolje predviđa SARS1 virus na test skupu. Odziv¹ za SARS1 na test skupu iznosi 66,67% (dok je on u slučajevima K-najbližih suseda iznosio 33,33%). Preciznost² za SARS1 na test skupu idalje iznosi 100%, kao i u slučajevima K-najbližih suseda. Idalje jedna instanca MERS soja biva pogrešno klasifikovana kao SARS2.

4.2 Modeli klasifikacije po tipu proteina

U narednim podpoglavlјima biće prikazani modeli klasifikacije po tipu proteina, koristeći različite 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

¹ Odziv se računa kao udeo tačno klasifikovanih instanci određene klase u ukupnom broj instanci te klase

² Preciznost se računa kao udeo tačno klasifikovanih instanci određene klase u ukupnom broju instanci koje su svrstane u tu klasu

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*

Stvarne klase	Predviđene klase																		
	ORF1a polyprotein	surface glycoprotein	ORF1ab polyprotein	ORF3a protein	ORF4b protein	ORF5 protein	envelope protein	nucleocapsid phosphoprotein	membrane glycoprotein	ORF8b protein	ORF4a protein	ORF3 protein	ORF7a protein	ORF6 protein	ORF8 protein	ORF7b protein	ORF1b polyprotein	ORF10 protein	
ORF1a polyprotein	-1371	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
surface glycoprotein	0	550	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF1ab polyprotein	2	0	1897	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF3a protein	0	0	0	73	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF4b protein	0	0	0	0	41	1	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF5 protein	0	0	0	0	1	34	0	0	0	0	0	0	0	0	0	0	0	0	0
envelope protein	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0
nucleocapsid phosphoprotein	0	0	0	0	0	0	0	211	0	0	0	0	0	0	0	0	0	0	0
membrane glycoprotein	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	0	0
ORF8b protein	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0
ORF4a protein	0	0	0	0	0	0	0	0	0	0	21	1	0	0	0	0	0	0	0
ORF3 protein	0	0	0	0	0	0	0	0	0	0	3	31	0	0	0	0	0	0	0
ORF7a protein	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0
ORF6 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
ORF8 protein	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	24	0	0	0
ORF7b protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0
ORF1b polyprotein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
ORF10 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Tabela 12 Matrica konfuzije Modela 5 na test skupu

Stvarne klase	Predviđene klase																	
	ORF1a polyprotein	surface glycoprotein	ORF1ab polyprotein	ORF3a protein	ORF4b protein	ORF5 protein	envelope protein	nucleocapsid phosphoprotein	membrane glycoprotein	ORF8b protein	ORF4a protein	ORF3 protein	ORF7a protein	ORF6 protein	ORF8 protein	ORF7b protein	ORF1b polyprotein	ORF10 protein
ORF1a polyprotein	2782	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
surface glycoprotein	0	1116	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF1ab polyprotein	1	0	3855	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF3a protein	0	0	0	148	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF4b protein	0	0	0	0	84	2	0	0	0	0	0	0	0	0	0	0	0	0
ORF5 protein	0	0	0	0	1	66	0	0	1	0	0	1	0	0	0	0	0	0
envelope protein	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0
nucleocapsid phosphoprotein	0	0	0	0	0	0	0	429	0	0	0	0	0	0	0	0	0	0
membrane glycoprotein	0	0	0	0	0	0	0	0	91	0	0	0	0	0	0	0	0	0
ORF8b protein	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0
ORF4a protein	0	0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0
ORF3 protein	0	0	0	0	0	0	0	0	0	0	1	69	0	0	0	0	0	0
ORF7a protein	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0
ORF6 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0
ORF8 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	1
ORF7b protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0
ORF1b polyprotein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
ORF10 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

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*

Stvarne klase	Predviđene klase																		
	ORF1a polyprotein	surface glycoprotein	ORF1ab polyprotein	ORF3a protein	ORF4b protein	ORF5 protein	envelope protein	nucleocapsid phosphoprotein	membrane glycoprotein	ORF8b protein	ORF4a protein	ORF3 protein	ORF7a protein	ORF6 protein	ORF8 protein	ORF7b protein	ORF1b polyprotein	ORF10 protein	
ORF1a polyprotein	1351	0	10	0	0	0	0	0	0	0	0	0	0	0	0	2	0	8	
surface glycoprotein	0	534	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	15	
ORF1ab polyprotein	30	0	1859	0	0	0	0	0	0	0	0	0	0	0	4	0	6	0	0
ORF3a protein	0	0	0	72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
ORF4b protein	0	0	0	0	39	1	0	0	0	0	0	0	0	0	0	0	0	0	2
ORF5 protein	0	0	0	0	2	32	0	0	0	0	0	0	0	0	0	0	0	0	1
envelope protein	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0
nucleocapsid phosphoprotein	0	0	0	0	0	0	0	204	0	0	0	0	0	0	0	2	0	5	
membrane glycoprotein	0	0	0	0	0	0	0	0	41	0	0	0	0	0	0	3	0	1	
ORF8b protein	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	3	0	1	
ORF4a protein	0	0	0	0	0	0	0	0	0	21	1	0	0	0	0	0	0	0	0
ORF3 protein	0	0	0	0	0	0	0	0	0	2	29	0	0	0	0	0	0	0	3
ORF7a protein	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0
ORF6 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
ORF8 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	3
ORF7b protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
ORF1b polyprotein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
ORF10 protein	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Tabela 14 Matrica konfuzije Modela 6 na test skupu

Stvarne Klase	ORF1a polyprotein	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	
	surface glycoprotein	0	1091	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	
	ORF1ab polyprotein	34	0	3817	0	0	0	0	0	0	0	0	0	0	0	0	4	0	1	
	ORF3a protein	0	0	0	148	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	ORF4b protein	0	0	0	0	84	2	0	0	0	0	0	0	0	0	0	0	0	0	
	ORF5 protein	0	0	0	0	1	68	0	0	0	0	0	0	0	0	0	0	0	0	
	envelope protein	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	
nucleocapsid phosphoprotein	0	0	0	0	0	0	0	423	0	0	0	0	0	0	0	0	6	0	0	
membrane glycoprotein	0	0	0	0	0	0	0	0	84	0	0	0	0	0	0	0	7	0	0	
	ORF8b protein	0	0	0	0	0	0	0	0	36	0	0	0	0	0	0	0	3	0	0
	ORF4a protein	0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	0
	ORF3 protein	0	0	0	0	0	0	0	0	0	0	70	0	0	0	0	0	0	0	0
	ORF7a protein	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0
	ORF6 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0
	ORF8 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	0	0	0
	ORF7b protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0
ORF1b polyprotein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	
ORF10 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
Predviđene klase																				

Tabela 15 Matrica konfuzije Modela 6 na trening skupu

Možemo primetiti lošu preciznost pri klasifikaciji manje zastupljenih klasa ORF7b (na trening skupu iznosi 12%, a na test 17,6%) i ORF10 (na trening skupu iznosi 6,25%, a na test 0) u odnosu na model 5 (koji ima 100% preciznost izuzev za ORF10 na trening skupu, kada iznosi 50%). Takođe, jedina instanca ORF10 proteina na test skupu ne biva prepoznata, za razliku od modela 5.

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`

Stvarne klase	Predviđene klase																		
	ORF1a polyprotein	surface glycoprotein	ORF1ab polyprotein	ORF3a protein	ORF4b protein	ORF5 protein	envelope protein	nucleocapsid phosphoprotein	membrane glycoprotein	ORF8b protein	ORF4a protein	ORF3 protein	ORF7a protein	ORF6 protein	ORF8 protein	ORF7b protein	ORF1b polyprotein	ORF10 protein	
ORF1a polyprotein	25	0	1346	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
surface glycoprotein	0	550	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF1ab polyprotein	68	0	1831	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF3a protein	0	0	0	73	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF4b protein	0	0	0	0	41	1	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF5 protein	0	0	0	0	1	34	0	0	0	0	0	0	0	0	0	0	0	0	0
envelope protein	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0
nucleocapsid phosphoprotein	0	0	0	0	0	0	0	211	0	0	0	0	0	0	0	0	0	0	0
membrane glycoprotein	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	0	0
ORF8b protein	0	0	0	0	0	1	0	0	0	19	0	0	0	0	0	0	0	0	0
ORF4a protein	0	0	0	0	0	0	0	0	0	0	21	1	0	0	0	0	0	0	0
ORF3 protein	0	0	0	0	0	0	0	0	0	0	3	31	0	0	0	0	0	0	0
ORF7a protein	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0
ORF6 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
ORF8 protein	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0
ORF7b protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0
ORF1b polyprotein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
ORF10 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabela 16 Matrica konfuzije Modela 7 na test skupu

Stvarne klase	ORF1a polyprotein	0	2687	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	surface glycoprotein	0	1115	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	ORF1ab polyprotein	70	0	3786	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	ORF3a protein	0	0	0	148	0	0	0	0	0	0	0	0	0	0	0	0	0
	ORF4b protein	0	0	0	0	84	2	0	0	0	0	0	0	0	0	0	0	0
	ORF5 protein	0	0	0	1	1	67	0	0	0	0	0	0	0	0	0	0	0
	envelope protein	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0
nucleocapsid phosphoprotein	0	0	0	0	0	0	0	429	0	0	0	0	0	0	0	0	0	0
membrane glycoprotein	0	1	0	0	0	0	0	0	90	0	0	0	0	0	0	0	0	0
ORF8b protein	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0
ORF4a protein	0	0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0
ORF3 protein	0	0	0	0	0	0	0	0	0	0	1	69	0	0	0	0	0	0
ORF7a protein	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0
ORF6 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0
ORF8 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	0	0
ORF7b protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0
ORF1b polyprotein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
ORF10 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	ORF1a polyprotein																	
	surface glycoprotein																	
	ORF1ab polyprotein																	
	ORF3a protein																	
	ORF4b protein																	
	ORF5 protein																	
	envelope protein																	
	nucleocapsid phosphoprotein																	
	membrane glycoprotein																	
	ORF8b protein																	
	ORF4a protein																	
	ORF3 protein																	
	ORF7a protein																	
	ORF6 protein																	
	ORF8 protein																	
	ORF7b protein																	
	ORF1b polyprotein																	
	ORF10 protein																	

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*

Stvarne klase	Predviđene klase																		
	ORF1a polyprotein	0	1343	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
surface glycoprotein	0	550	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF1ab polyprotein	68	0	1831	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF3a protein	0	0	0	73	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF4b protein	0	0	0	0	41	1	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF5 protein	0	0	0	0	1	34	0	0	0	0	0	0	0	0	0	0	0	0	0
envelope protein	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0
nucleocapsid phosphoprotein	0	0	0	0	0	0	0	211	0	0	0	0	0	0	0	0	0	0	0
membrane glycoprotein	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	0	0
ORF8b protein	0	0	0	0	0	1	0	0	0	19	0	0	0	0	0	0	0	0	0
ORF4a protein	0	0	0	0	0	0	0	0	0	0	21	1	0	0	0	0	0	0	0
ORF3 protein	0	0	0	0	0	0	0	0	0	0	3	31	0	0	0	0	0	0	0
ORF7a protein	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0
ORF6 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
ORF8 protein	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	24	0	0	0
ORF7b protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0
ORF1b polyprotein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
ORF10 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabela 18 Matrica konfuzije Modela 8 na test skupu

Stvarne klase	ORF1a polyprotein	0	2686	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	surface glycoprotein	0	1114	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
	ORF1ab polyprotein	73	0	3783	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	ORF3a protein	0	0	0	148	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	ORF4b protein	0	0	0	0	84	2	0	0	0	0	0	0	0	0	0	0	0	0	0
	ORF5 protein	0	0	0	0	1	68	0	0	0	0	0	0	0	0	0	0	0	0	0
	envelope protein	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0
nucleocapsid phosphoprotein	0	0	0	0	0	0	0	0	429	0	0	0	0	0	0	0	0	0	0	
membrane glycoprotein	0	0	0	0	0	0	0	0	0	90	0	0	0	0	0	0	0	0	1	0
ORF8b protein	0	0	0	0	0	0	0	0	1	0	38	0	0	0	0	0	0	0	0	0
ORF4a protein	0	0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	0
ORF3 protein	0	1	0	0	0	0	0	0	0	0	0	1	68	0	0	0	0	0	0	0
ORF7a protein	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0
ORF6 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0
ORF8 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	0	0	0
ORF7b protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0
ORF1b polyprotein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
ORF10 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	ORF1a polyprotein																			
	surface glycoprotein																			
	ORF1ab polyprotein																			
	ORF3a protein																			
	ORF4b protein																			
	ORF5 protein																			
	envelope protein																			
	nucleocapsid phosphoprotein																			
	membrane glycoprotein																			
	ORF8b protein																			
	ORF4a protein																			
	ORF3 protein																			
	ORF7a protein																			
	ORF6 protein																			
	ORF8 protein																			
	ORF7b protein																			
	ORF1b polyprotein																			
	ORF10 protein																			

Tabela 19 Matrica konfuzije Modela 8 na trening skupu

4.2.5 Model zasnovan na metodu potpornih vektora

Ovaj model primjenjuje metodu potpornih vektora (SVM – Support Vector Machine) iz biblioteke scikit-learn[9] za klasifikaciju tipova proteina.

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

Ocena modela:

Dobijene su sledeće vrednosti tačnosti modela na test skupu:

- **Test skup:** 0.9986

U nastavku je prikazana matrica konfuzije Modela 2 na test skupu. Ova matrica je sačuvana na putanji *results/heatmaps/svm/protein*.

Stvarne klase	ORF1a polyprotein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Predviđene klase	ORF1a polyprotein	1371	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF1ab polyprotein	0	1899	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
surface glycoprotein	0	0	550	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nucleocapsid phosphoprotein	0	0	0	211	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
membrane glycoprotein	0	0	0	0	45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF8b protein	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF3 protein	0	0	0	0	0	0	32	0	0	0	0	0	0	2	0	0	0	0	0	0
ORF4b protein	0	0	0	0	0	0	0	41	0	1	0	0	0	0	0	0	0	0	0	0
ORF8 protein	0	0	0	0	0	0	1	0	24	0	0	0	0	0	0	0	0	0	0	0
ORF5 protein	0	0	0	0	0	0	0	0	0	35	0	0	0	0	0	0	0	0	0	0
ORF3a protein	0	0	0	0	0	0	0	0	0	0	73	0	0	0	0	0	0	0	0	0
ORF4a protein	0	0	0	0	0	0	1	0	0	0	0	21	0	0	0	0	0	0	0	0
envelope protein	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0
ORF6 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
ORF7b protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
ORF7a protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0
ORF10 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
ORF1b polyprotein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2

Možemo primetiti da se ne dobija značajno poboljšanje u odnosu na algoritam K-najbližih suseda.

4.3 Modeli klasifikacije po soju SARS2 virusa

U narednim podpoglavlјima 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*

	Omicron	Iota	Alpha	Epsilon	Kappa	Gamma	Eta	Delta	Beta	Lambda	Mu	Zeta	Theta	
Stvarne klase	476	3	3	1	0	4	0	4	0	1	0	3	0	
Iota	1	488	0	1	0	0	0	2	3	0	0	0	0	
Alpha	0	4	482	1	0	0	1	0	0	0	0	7	0	
Epsilon	1	0	0	484	0	1	1	2	6	0	0	0	0	
Kappa	0	0	2	0	47	0	0	3	0	0	0	0	0	
Gamma	5	3	6	2	0	478	0	0	0	0	0	0	1	
Eta	0	2	0	0	0	0	197	0	2	0	0	6	0	
Delta	3	0	5	2	5	1	0	479	0	0	0	0	0	
Beta	2	6	2	4	0	0	0	3	176	0	0	0	0	
Lambda	0	0	1	1	0	0	0	3	0	127	0	3	0	
Mu	0	1	0	1	0	0	0	0	2	0	18	0	0	
Zeta	0	0	1	0	1	0	0	0	0	0	152	0	0	
Theta	0	0	0	0	0	0	0	2	0	0	0	0	5	

Tabela 20 Matrica konfuzije Modela 9 na test skupu

	Omicron	Iota	Alpha	Epsilon	Kappa	Gamma	Eta	Delta	Beta	Lambda	Mu	Zeta	Theta	
Stvarne klase	983	1	5	7	2	0	0	3	2	1	0	1	0	
Iota	2	996	2	1	0	0	1	1	1	1	0	0	0	
Alpha	0	4	990	1	0	0	0	0	1	0	0	9	0	
Epsilon	1	1	1	986	0	0	1	9	5	0	1	0	0	
Kappa	0	1	1	0	99	0	0	4	0	0	0	0	0	
Gamma	5	2	1	3	0	991	1	1	1	0	0	0	0	
Eta	2	2	3	0	0	0	403	4	1	0	0	6	0	
Delta	4	3	1	0	2	0	0	992	2	1	0	0	0	
Beta	2	2	1	3	1	0	0	0	384	0	0	0	0	
Lambda	0	2	0	1	0	0	0	4	1	265	0	2	0	
Mu	0	0	0	3	0	0	0	1	2	0	37	0	0	
Zeta	0	0	2	0	0	0	0	3	1	0	0	307	0	
Theta	1	0	0	0	0	0	1	0	0	0	0	1	10	
Predviđene klase	Omicron	Iota	Alpha	Epsilon	Kappa	Gamma	Eta	Delta	Beta	Lambda	Mu	Zeta	Theta	

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 *source/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*

Stvarne klase	Omicron	Iota	Alpha	Epsilon	Kappa	Gamma	Eta	Delta	Beta	Lambda	Mu	Zeta	Theta
Predviđene klase	Omicron	Iota	Alpha	Epsilon	Kappa	Gamma	Eta	Delta	Beta	Lambda	Mu	Zeta	Theta
Omicron	480	1	10	0	0	0	0	4	0	0	0	0	0
Iota	0	492	1	1	0	1	0	0	0	0	0	0	0
Alpha	1	2	489	0	0	0	0	1	2	0	0	0	0
Epsilon	0	0	3	488	0	0	0	2	2	0	0	0	0
Kappa	0	0	2	1	46	0	0	3	0	0	0	0	0
Gamma	0	2	7	2	0	476	2	0	6	0	0	0	0
Eta	0	1	7	0	0	0	198	1	0	0	0	0	0
Delta	0	0	9	0	5	0	0	480	1	0	0	0	0
Beta	2	0	2	2	0	0	0	4	183	0	0	0	0
Lambda	0	1	4	0	0	0	0	3	0	127	0	0	0
Mu	0	2	0	1	0	0	0	0	2	0	17	0	0
Zeta	0	0	3	0	0	0	0	0	1	0	0	150	0
Theta	0	0	1	0	0	0	0	1	0	0	0	0	5

Tabela 22 Matrica konfuzije Modela 10 na test skupu

	Omicron	Iota	Alpha	Epsilon	Kappa	Gamma	Eta	Delta	Beta	Lambda	Mu	Zeta	Theta	
Stvarne klase	987	0	11	2	0	0	0	3	1	1	0	0	0	
Iota	1	995	6	1	0	2	0	0	0	0	0	0	0	
Alpha	1	1	1002	0	0	0	0	0	1	0	0	0	0	
Epsilon	0	2	4	995	0	0	0	3	1	0	0	0	0	
Kappa	0	0	2	0	101	0	0	2	0	0	0	0	0	
Gamma	0	1	1	2	0	1001	0	0	0	0	0	0	0	
Eta	0	1	10	0	0	0	405	4	1	0	0	0	0	
Delta	1	1	3	0	1	0	0	998	1	0	0	0	0	
Beta	0	1	1	3	0	0	0	3	385	0	0	0	0	
Lambda	0	0	6	0	0	0	0	3	1	265	0	0	0	
Mu	0	1	1	3	0	0	0	1	1	0	36	0	0	
Zeta	0	0	8	1	0	0	0	5	1	0	0	298	0	
Theta	0	0	2	0	0	1	0	0	0	0	0	0	10	
Predviđene klase	Omicron	Iota	Alpha	Epsilon	Kappa	Gamma	Eta	Delta	Beta	Lambda	Mu	Zeta	Theta	

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 *source/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*.

	Omicron	Iota	Alpha	Epsilon	Kappa	Gamma	Eta	Delta	Beta	Lambda	Mu	Zeta	Theta
Stvarne klase	65	3	417	2	0	4	0	3	0	1	0	0	0
	Omicron	Iota	Alpha	Epsilon	Kappa	Gamma	Eta	Delta	Beta	Lambda	Mu	Zeta	Theta
Omicron	65	3	417	2	0	4	0	3	0	1	0	0	0
Iota	0	65	409	1	0	2	0	18	0	0	0	0	0
Alpha	0	0	488	1	0	2	0	3	1	0	0	0	0
Epsilon	0	1	389	81	0	0	0	24	0	0	0	0	0
Kappa	0	0	30	14	0	0	0	8	0	0	0	0	0
Gamma	0	1	399	1	0	70	0	22	2	0	0	0	0
Eta	3	3	151	1	0	0	48	0	1	0	0	0	0
Delta	0	1	398	2	0	1	0	91	2	0	0	0	0
Beta	0	3	127	27	0	0	0	20	15	1	0	0	0
Lambda	0	13	84	0	0	1	0	4	2	31	0	0	0
Mu	0	0	14	5	0	0	0	3	0	0	0	0	0
Zeta	0	1	114	19	0	0	0	19	0	0	0	1	0
Theta	0	0	3	2	0	0	0	2	0	0	0	0	0

Tabela 24 Matrica konfuzije Modela 11 na test skupu

	Omicron	Iota	Alpha	Epsilon	Kappa	Gamma	Eta	Delta	Beta	Lambda	Mu	Zeta	Theta	
Stvarne klase	122	3	866	6	0	2	1	4	0	1	0	0	0	
Iota	1	143	817	1	0	3	0	38	1	1	0	0	0	
Alpha	0	0	997	1	0	0	0	5	2	0	0	0	0	
Epsilon	0	1	798	155	0	0	0	48	3	0	0	0	0	
Kappa	0	1	62	22	2	2	0	15	1	0	0	0	0	
Gamma	0	3	797	6	0	147	0	49	3	0	0	0	0	
Eta	1	0	296	2	0	0	118	3	1	0	0	0	0	
Delta	0	1	811	0	0	0	0	190	3	0	0	0	0	
Beta	0	2	258	47	0	0	0	31	54	1	0	0	0	
Lambda	0	19	193	1	0	1	0	9	1	51	0	0	0	
Mu	0	0	23	14	0	0	0	6	0	0	0	0	0	
Zeta	0	3	244	29	0	0	0	23	2	1	0	11	0	
Theta	0	0	9	2	0	0	0	2	0	0	0	0	0	

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*

Stvarne klase	Omicron	Iota	Alpha	Epsilon	Kappa	Gamma	Eta	Delta	Beta	Lambda	Mu	Zeta	Theta
Predviđene klase	Omicron	Iota	Alpha	Epsilon	Kappa	Gamma	Eta	Delta	Beta	Lambda	Mu	Zeta	Theta
Omicron	65	3	417	4	0	2	0	3	0	1	0	0	0
Iota	0	65	409	1	0	1	0	19	0	0	0	0	0
Alpha	0	0	488	1	0	2	0	3	1	0	0	0	0
Epsilon	0	0	390	81	0	0	0	24	0	0	0	0	0
Kappa	0	0	30	14	0	0	0	8	0	0	0	0	0
Gamma	0	1	399	1	0	71	0	22	1	0	0	0	0
Eta	3	3	151	1	0	0	47	1	1	0	0	0	0
Delta	0	0	397	2	0	1	0	95	0	0	0	0	0
Beta	0	3	129	27	0	0	0	21	13	0	0	0	0
Lambda	0	12	84	0	0	1	0	5	1	32	0	0	0
Mu	0	0	14	5	0	0	0	3	0	0	0	0	0
Zeta	0	1	114	19	0	0	0	19	0	0	0	1	0
Theta	0	0	3	2	0	0	0	2	0	0	0	0	0

Tabela 26 Matrica konfuzije Modela 12 na test skupu

	Omicron	Iota	Alpha	Epsilon	Kappa	Gamma	Eta	Delta	Beta	Lambda	Mu	Zeta	Theta	
Stvarne klase	123	3	866	6	0	2	1	4	0	0	0	0	0	0
Iota	2	144	817	1	0	2	0	38	0	1	0	0	0	0
Alpha	0	0	998	1	0	0	0	5	1	0	0	0	0	0
Epsilon	0	0	798	160	0	0	0	47	0	0	0	0	0	0
Kappa	0	1	62	22	2	2	0	15	1	0	0	0	0	0
Gamma	0	3	797	5	0	149	0	49	2	0	0	0	0	0
Eta	2	0	296	2	0	0	116	5	0	0	0	0	0	0
Delta	0	0	810	0	0	0	0	194	1	0	0	0	0	0
Beta	0	2	260	48	0	0	0	32	51	0	0	0	0	0
Lambda	0	19	194	1	0	1	0	9	1	50	0	0	0	0
Mu	0	0	23	14	0	0	0	6	0	0	0	0	0	0
Zeta	0	3	245	29	0	0	0	23	1	0	0	12	0	0
Theta	0	0	10	2	0	0	0	1	0	0	0	0	0	0
Predviđene klase	Omicron	Iota	Alpha	Epsilon	Kappa	Gamma	Eta	Delta	Beta	Lambda	Mu	Zeta	Theta	

Tabela 27 Matrica konfuzije Modela 12 na trening skupu

4.3.5 Model zasnovan na metodu potpornih vektora

Ovaj model primenjuje metodu potpornih vektora (SVM – Support Vector Machine) iz biblioteke scikit-learn[9] za klasifikaciju sojeva SARS2 virusa.

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

Ocena modela:

Dobijene su sledeće vrednosti tačnosti modela na test skupu:

- **Test skup:** 0.971

U nastavku je prikazana matrica konfuzije Modela 3 na test skupu. Ova matrica je sačuvana na putanji *results/heatmaps/svm/sars2*.

	Zeta	Omicron	Epsilon	Iota	Alpha	Eta	Gamma	Lambda	Delta	Kappa	Beta	Mu	Theta	
Stvarne klase	152	0	0	1	0	0	0	1	0	0	0	0	0	0
Predviđene klase	Zeta	Omicron	Epsilon	Iota	Alpha	Eta	Gamma	Lambda	Delta	Kappa	Beta	Mu	Theta	
Zeta	152	0	0	1	0	0	0	1	0	0	0	0	0	0
Omicron	2	482	1	4	3	0	0	0	3	0	0	0	0	0
Epsilon	1	0	487	1	0	0	0	0	4	0	2	0	0	0
Iota	0	1	1	493	0	0	0	0	0	0	0	0	0	0
Alpha	6	0	1	0	484	0	0	2	1	0	1	0	0	0
Eta	6	0	1	1	0	197	1	0	1	0	0	0	0	0
Gamma	0	0	2	3	3	0	478	3	0	0	6	0	0	0
Lambda	3	0	2	0	0	0	0	127	3	0	0	0	0	0
Delta	0	0	2	0	2	0	2	0	482	6	1	0	0	0
Kappa	0	0	0	0	2	0	0	0	3	47	0	0	0	0
Beta	0	2	0	3	0	0	0	3	3	0	182	0	0	0
Mu	0	0	2	1	0	0	0	0	0	0	2	17	0	0
Theta	0	0	0	0	0	0	1	0	1	0	0	0	5	0

Možemo primetiti da nema značajnih poboljšanja u odnosu na slučaj K-najbližih suseda.

4.4 Optimizacija modela

Izvršena je optimizacija hiperparametra k modela, zasnovanih na algoritmu k najbližih suseda. Optimizacija je vršena metodom unakrsne provere. Unakrsna provera je vršena za $k = 1, 2, \dots, 5$. Originalni skup podataka se delio na 3 podskupa (skoro) jednake veličine i svaki od podskupova je korišćen kao validacioni skup. Zbog velikih razlika u broju instanci klasa, korišćena je prosečna F1 mera kao metrika za ocenu kvaliteta modela. Prosečna F1 mera se računa kao prosek F1 mera svake od klase. F1 mera pojedinačne klase se računa kao harmonijska sredina preciznosti i odziva. Rastojanje se računa kao kombinacija 5-adične i 2-adične distance sa adaptivnim odsecanjem (opisanom u sekciji [3.1](#)).

Izvorni kod optimizacije modela vršene metodom unakrsne provere se nalazi u datoteci na putanji `source/knn_cross_validation.ipynb`. Funkcija koja računa rastojanje je `five_adic_distance_from_matrix(x, y)`. Argumenti x i y su liste koje sadrže pozicije sekvenci u originalnom [skupu podataka](#), na poslednjem mestu. Iz matrice rastojanja (na putanji

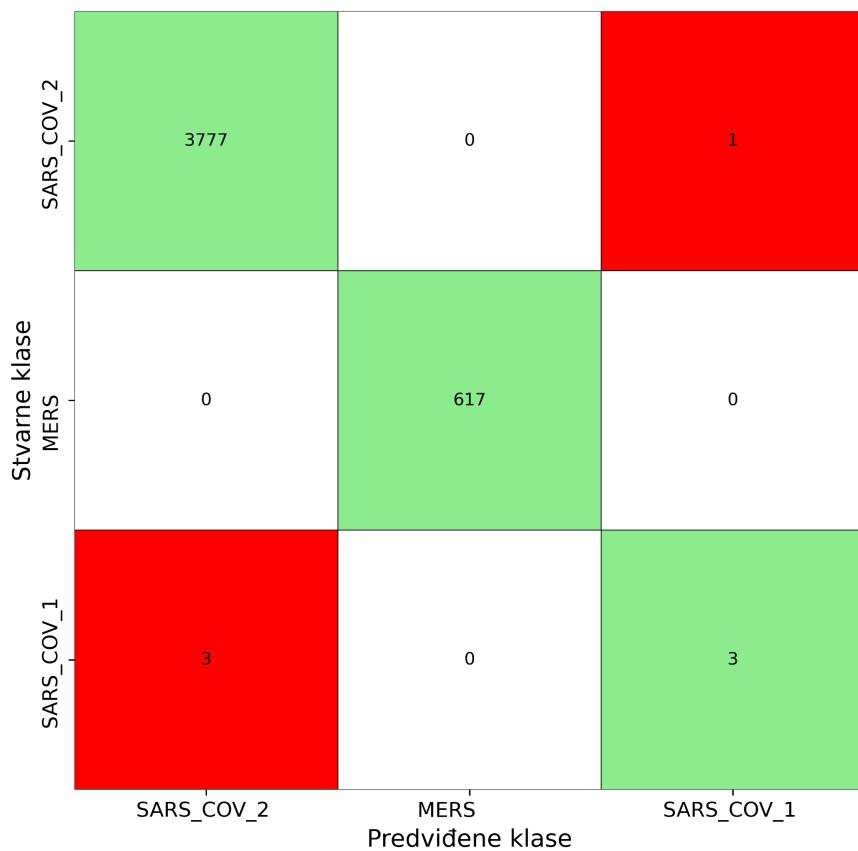
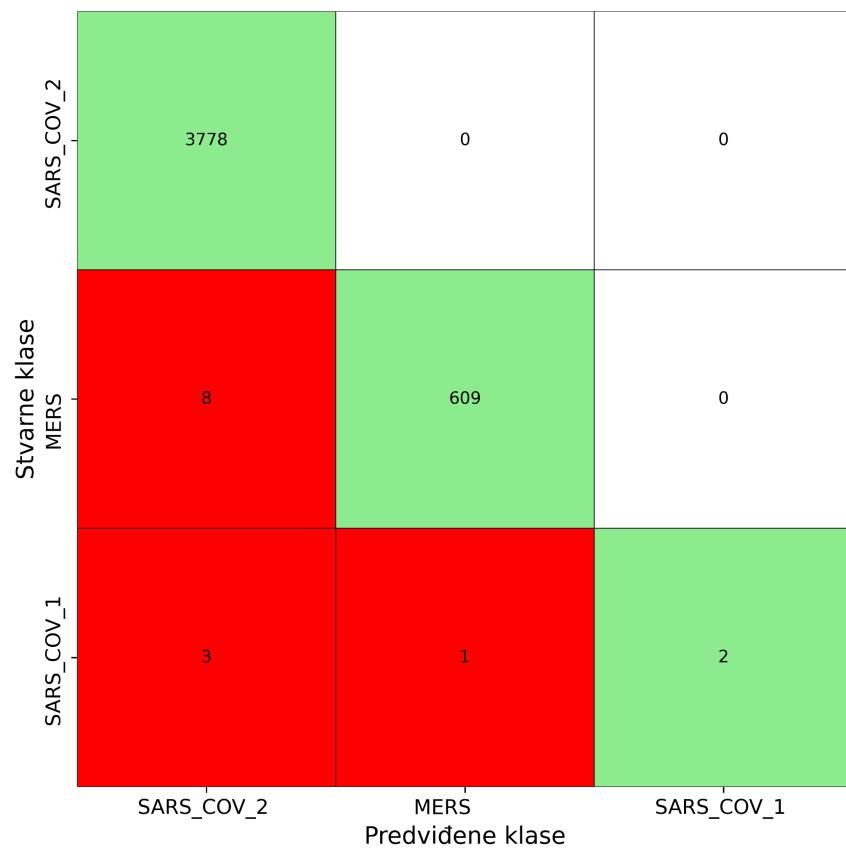
data/full_distances_matrix.7z) se na osnovu njihovih pozicija izvlači podatak o medjusobnom rastojanju i vraća. Prosečna F1 mera se računa pomoću proizvoljno definisane metrike (*custom_f1* promenljiva u kodu), koja je napravljena korišćenjem *make_scorer* funkcije iz modula *sklearn.metrics*[7]. Metrika se definiše preko funkcije *custom_f1_loss(y_true, y_pred)*, koja vraća rezultat funkcije *f1_score* iz modula *sklearn.metrics*[6]. Argument *y_true* predstavlja listu ispravnih klasa sekvenci na validacionom ili trening skupu, a *y_pred* listu predviđenih. Bočni efekat jeste poziv funkcije *visualize_confusion_matrix(cm, labels, save_path="")*, koji čuva matricu konfuzije (parametar *cm*) kao topotnu mapu na putanji *results/heatmaps/cross_validation*. Unakrsna provera se vrši korišćenjem klase *GridSearchCV* iz modula *sklearn.model_selection*[8].

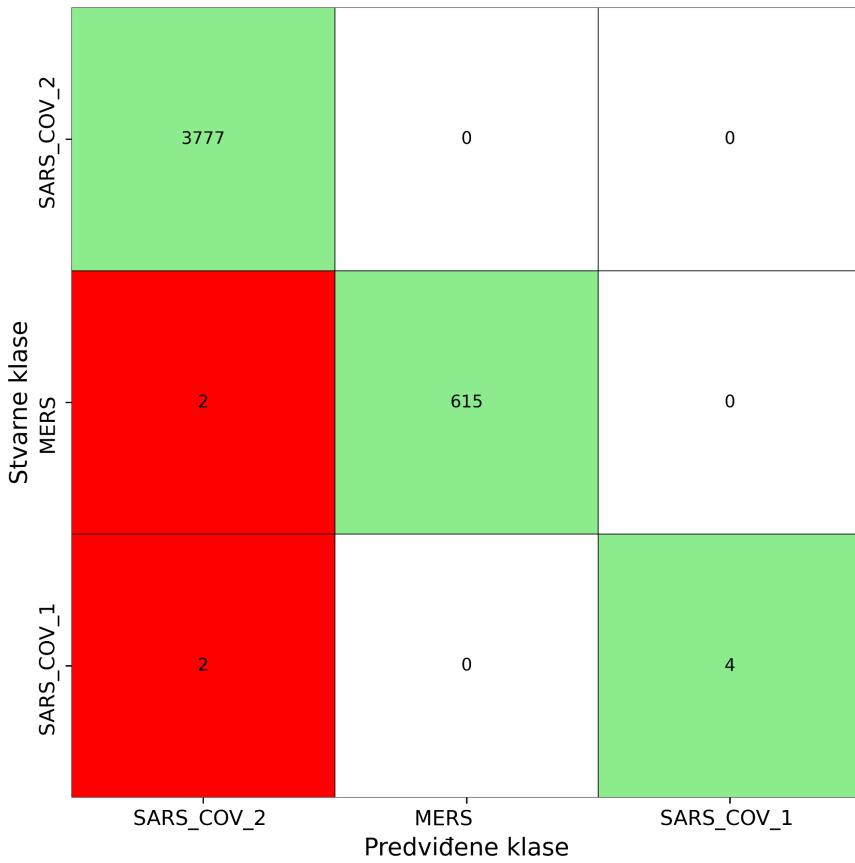
4.4.1. Optimizacija modela za klasifikaciju po tipu virusa

U nastavku su prikazani rezultati prosečne F1 mere pri klasifikaciji po tipu virusa na trening i validacionim skupovima.

```
n_neighors=1;; score=(train=1.000, test=0.830)
n_neighors=1;; score=(train=1.000, test=0.866)
n_neighors=1;; score=(train=1.000, test=0.933)
n_neighors=2;; score=(train=0.985, test=0.759)
n_neighors=2;; score=(train=0.969, test=0.800)
n_neighors=2;; score=(train=0.938, test=0.781)
n_neighors=3;; score=(train=0.912, test=0.663)
n_neighors=3;; score=(train=0.888, test=0.760)
n_neighors=3;; score=(train=0.842, test=0.665) n_neighors=4;; score=(train=0.912, test=0.663)
n_neighors=4;; score=(train=0.888, test=0.760)
n_neighors=4;; score=(train=0.783, test=0.665)
n_neighors=5;; score=(train=0.862, test=0.663)
n_neighors=5;; score=(train=0.832, test=0.760)
n_neighors=5;; score=(train=0.665, test=0.663)
```

Premda slučaj $k=1$ kod algoritma k najблиžih suseda predstavlja preprilagođavanje (jer se pri treniranju u obzir uzima samo distanca instance od same sebe), primetimo da ova vrednost ipak daje najbolje rezultate i na validacionom skupu. U nastavku su prikazane matrice konfuzije za slučaj $k=1$ na validacionim skupovima (nalaze se na putanji *results/heatmaps/cross_validation/virus*).





Primećujemo da je modelu najteže da prepozna SARS_COV_1 soj. Na ova tri validaciona skupa se pokazalo da će model u 50% slučajeva SARS_COV_1 soj klasifikovati tačno (odziv SARS_COV_1 klasifikacije je 50%). Kada model klasificiše neku sekvencu kao SARS_COV_1 u 90% slučajeva je to ispravno (preciznost SARS_COV_1 klasifikacije je 90%). SARS_COV_1 se najčešće pogrešno klasificiše kao najbrojniji SARS_COV_2 soj, što očigledno pokazuje njihovu sličnost. MERS i SARS_COV_2 se prilično uspešno klasificuju.

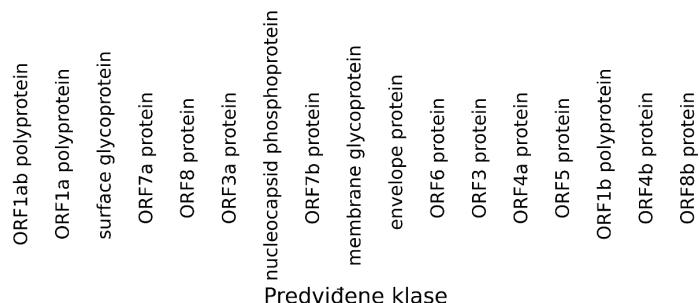
4.4.2. Optimizacija modela za klasifikaciju po tipu proteina

U nastavku su prikazani rezultati prosečne F1 mere pri klasifikaciji po tipu proteina na trening i validacionim skupovima.

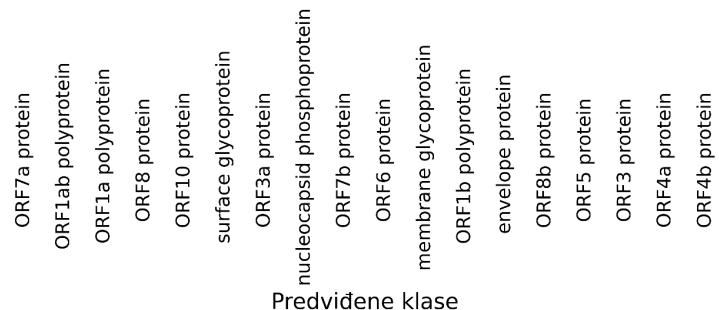
```
n_neighors=1;; score=(train=0.997, test=0.960)
n_neighors=1;; score=(train=0.999, test=0.987) n_neighors=4;; score=(train=0.992, test=0.997)
n_neighors=1;; score=(train=0.999, test=0.972) n_neighors=4;; score=(train=0.929, test=0.916)
n_neighors=2;; score=(train=0.980, test=0.950) n_neighors=4;; score=(train=0.931, test=0.925)
n_neighors=2;; score=(train=0.992, test=0.982) n_neighors=5;; score=(train=0.928, test=0.982)
n_neighors=2;; score=(train=0.974, test=0.965) n_neighors=5;; score=(train=0.927, test=0.916)
n_neighors=3;; score=(train=0.994, test=0.997) n_neighors=5;; score=(train=0.931, test=0.926)
n_neighors=3;; score=(train=0.975, test=0.980)
n_neighors=3;; score=(train=0.978, test=0.988)
```

Najbolji rezultati na validacionom skupu se dobiju za vrednost k = 3. U nastavku su prikazane matrice konfuzije za slučaj k=3 na validacionim skupovima (nalaze se na putanji *results/heatmaps/cross_validation/protein*).

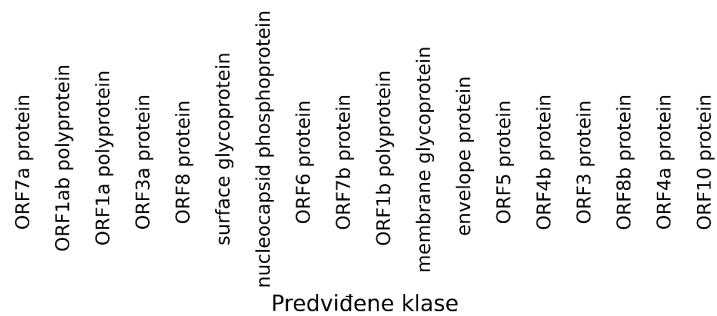
Stvarne klase	ORF1ab polyprotein	ORF1a polyprotein	surface glycoprotein	ORF7a protein	ORF8 protein	ORF3a protein	nucleocapsid phosphoprotein	ORF7b protein	membrane glycoprotein	envelope protein	ORF6 protein	ORF3 protein	ORF4a protein	ORF5 protein	ORF1b polyprotein	ORF4b protein	ORF8b protein	
ORF1ab polyprotein	1919	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF1a polyprotein	0	1384	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
surface glycoprotein	0	0	555	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF7a protein	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF8 protein	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF3a protein	0	0	0	0	0	74	0	0	0	0	0	0	0	0	0	0	0	0
nucleocapsid phosphoprotein	0	0	0	0	0	0	213	0	0	0	0	0	0	0	0	0	0	0
ORF7b protein	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
membrane glycoprotein	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	0	0
envelope protein	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0
ORF6 protein	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
ORF3 protein	0	0	0	0	0	0	0	0	0	0	0	34	0	0	0	0	0	0
ORF4a protein	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0
ORF5 protein	0	0	0	0	0	0	0	0	0	0	0	0	1	33	0	0	0	0
ORF1b polyprotein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
ORF4b protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	42	0	0
ORF8b protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20



Stvarne klase	ORF7a protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF1ab polyprotein	-	0	1918	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF1a polyprotein	-	0	0	1385	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF8 protein	-	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF10 protein	-	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
surface glycoprotein	-	0	0	0	0	0	555	0	0	0	0	0	0	0	0	0	0	0
ORF3a protein	-	0	0	0	0	0	0	73	0	0	0	0	0	0	0	0	0	0
nucleocapsid phosphoprotein	-	0	0	0	0	0	0	0	214	0	0	0	0	0	0	0	0	0
ORF7b protein	-	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
ORF6 protein	-	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
membrane glycoprotein	-	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0
ORF1b polyprotein	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
envelope protein	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF8b protein	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF5 protein	-	0	0	0	0	0	0	0	0	0	0	1	0	0	0	34	0	0
ORF3 protein	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	34	1	0
ORF4a protein	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0
ORF4b protein	-	0	0	0	0	0	0	0	0	0	0	0	1	0	3	1	4	34



Stvarne klase	ORF7a protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF1ab polyprotein	0	1917	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF1a polyprotein	0	0	1384	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF3a protein	0	0	0	74	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ORF8 protein	0	0	0	0	24	0	0	0	0	0	0	0	1	0	0	0	0	0
surface glycoprotein	0	0	0	0	0	556	0	0	0	0	0	0	0	0	0	0	0	0
nucleocapsid phosphoprotein	0	0	0	0	0	0	213	0	0	0	0	0	0	0	0	0	0	0
ORF6 protein	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
ORF7b protein	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
ORF1b polyprotein	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
membrane glycoprotein	0	0	0	0	0	0	0	0	0	46	0	0	0	0	0	0	0	0
envelope protein	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0
ORF5 protein	0	0	0	0	0	0	0	0	0	0	0	34	1	0	0	0	0	0
ORF4b protein	0	0	0	0	0	0	0	0	0	0	0	2	41	0	0	0	0	0
ORF3 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	34	0	1	0	0
ORF8b protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0
ORF4a protein	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	21	0	0
ORF10 protein	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1



Primetimo da po jedna instance ORF5 proteina biva pogrešno klasifikovana kao ORF4b, membranski glikoprotein i ORF4a. Budući da je ORF5 protein endoplazmatičnog retikuluma nije ni čudo da su pojedene instance slične proteinima membrane (koja ograničava retikulum) i jedra. Najčešće se javljaju greške pri klasifikacij ORF proteina i to se najčešće loše klasificuje ORF4b proteina. Odziv pri klasifikaciji ORF4b proteina je 91.4%. Sve klase sa jednocifrešnim brojem instanci se klasikuju sa preciznošću i odzivom 100%.

4.4.3. Optimizacija modela za klasifikaciju SARS 2 soja po Svetskoj zdravstvenoj organizaciji

U nastavku su prikazani rezultati prosečne F1 mere pri klasifikaciji Sars 2 soja po Svetskoj zdravstvenoj organizaciji na trening i validacionim skupovima.

```
n_neighbors=1;, score=(train=1.000, test=0.876)
n_neighbors=1;, score=(train=1.000, test=0.944)
n_neighbors=1;, score=(train=1.000, test=0.941)
n_neighbors=2;, score=(train=0.974, test=0.875)
n_neighbors=2;, score=(train=0.964, test=0.944)
n_neighbors=2;, score=(train=0.970, test=0.947)
n_neighbors=3;, score=(train=0.971, test=0.878)
n_neighbors=3;, score=(train=0.955, test=0.948)
n_neighbors=3;, score=(train=0.958, test=0.949)

n_neighbors=4;, score=(train=0.961, test=0.882)
n_neighbors=4;, score=(train=0.949, test=0.947)
n_neighbors=4;, score=(train=0.955, test=0.950)
n_neighbors=5;, score=(train=0.960, test=0.882)
n_neighbors=5;, score=(train=0.944, test=0.946)
n_neighbors=5;, score=(train=0.957, test=0.943)
```

Najbolji rezultati na validacionom skupu se dobiju za vrednost k = 4. U nastavku su prikazane matrice konfuzije za slučaj k = 4 na validacionim skupovima (nalaze se na putanji *results/heatmaps/cross_validation/who*).

	Alpha	Delta	Epsilon	Gamma	Iota	Omicron	Zeta	Kappa	Beta	Eta	Lambda	Mu	Theta	
Stvarne klase	482	4	1	1	1	0	0	0	4	7	0	0	0	
Alpha	482	4	1	1	1	0	0	0	4	7	0	0	0	
Delta	11	481	1	0	2	2	0	1	1	1	0	0	0	
Epsilon	1	3	490	0	2	1	0	0	0	2	0	1	0	
Gamma	2	1	3	487	0	4	0	0	2	0	0	1	0	
Iota	6	0	2	1	484	4	0	0	3	0	0	0	0	
Omicron	15	13	3	15	73	297	0	0	79	5	0	0	0	
Zeta	0	3	0	0	0	0	149	0	1	3	0	0	0	
Kappa	1	10	0	0	0	0	0	40	0	1	0	0	0	
Beta	1	0	1	0	1	0	0	0	192	0	0	1	0	
Eta	1	1	2	0	0	2	0	0	1	202	0	0	0	
Lambda	2	1	4	0	2	0	0	0	0	2	126	0	0	
Mu	1	1	2	0	2	0	0	0	1	0	0	14	0	
Theta	0	0	0	1	0	1	0	0	0	1	0	0	4	
Predviđene klase														

Stvarne klase	Alpha	Delta	Epsilon	Gamma	Iota	Omicron	Zeta	Eta	Kappa	Lambda	Beta	Mu	Theta	
Predviđene klase	Alpha	483	0	1	2	6	1	7	0	0	0	0	0	0
Alpha	483	0	1	2	6	1	7	0	0	0	0	0	0	0
Delta	0	487	2	3	2	5	0	0	0	1	0	0	0	0
Epsilon	0	8	488	0	0	2	0	0	0	0	2	0	0	0
Gamma	1	2	5	487	0	3	0	0	0	0	0	1	1	0
Iota	0	1	3	3	490	2	0	1	0	0	0	0	0	0
Omicron	0	2	2	1	1	490	3	0	0	0	0	1	0	0
Zeta	0	1	0	0	0	0	154	0	1	0	0	0	0	0
Eta	0	3	0	0	0	0	4	202	0	0	1	0	0	0
Kappa	1	1	0	0	0	0	0	0	50	0	0	0	0	0
Lambda	0	3	0	0	0	0	3	0	0	124	6	0	0	0
Beta	0	3	2	1	0	3	0	1	0	0	185	0	0	0
Mu	1	0	2	0	1	0	0	0	0	0	2	16	0	0
Theta	0	0	0	1	0	0	0	0	0	0	0	0	6	0

	Alpha	Delta	Epsilon	Gamma	Iota	Omicron	Zeta	Kappa	Mu	Lambda	Eta	Theta	Beta	
Stvarne klase	490	2	3	3	0	0	1	0	0	0	0	0	1	0
Predviđene klase	Alpha	Delta	Epsilon	Gamma	Iota	Omicron	Zeta	Kappa	Mu	Lambda	Eta	Theta	Beta	
Alpha	490	2	3	3	0	0	1	0	0	0	0	0	1	0
Delta	1	483	9	0	2	0	0	4	0	0	0	0	0	1
Epsilon	0	6	487	1	0	4	0	1	0	0	0	0	0	1
Gamma	1	0	1	496	0	2	0	0	0	0	0	0	0	0
Iota	1	0	0	4	493	1	0	0	0	0	1	0	0	0
Omicron	2	1	5	0	4	479	2	0	0	2	5	0	0	0
Zeta	3	3	2	0	0	0	147	0	0	0	0	0	0	0
Kappa	1	2	2	0	0	0	0	48	0	0	0	0	0	0
Mu	0	0	2	0	0	0	0	0	19	0	0	0	0	1
Lambda	5	2	0	0	1	0	0	0	0	128	0	0	0	1
Eta	8	1	0	2	3	4	0	0	0	0	190	0	0	1
Theta	0	1	0	0	0	0	0	0	0	0	0	5	0	0
Beta	2	4	8	0	4	0	0	1	0	0	0	0	0	176

Možemo primetiti da Teta soj biva najlošije prepoznat, ukupan odziv iznosi 75%, dok na prvom skupu iznosi oko 57%. Mu soj na prvom validacionom skupu ima odziv 75,3%, a 66,67% na prvom validacionom skupu. Omikron soj zatim biva najlošije prepoznat, iako je ukupan odziv na sva 3 skupa 84,4%, dok na prvom skupu on iznosi samo 59,4%. Najčešće se Omikron pogrešno klasificiše kao Jota (73) i Beta (79), u slučaju prvog validacionog skupa.

5. Zaključak

Poređenjem modela, sa [adaptivnim odsecanjem](#), najčešće se uočava bolja tačnost na trening i test skupu modela zasnovanih na p-adičnoj metrici u odnosu na model zasnovan na Hamingovom rastojanju, premda je rezlika izuzetno mala. U slučaju klasifikacije po tipu virusa razlika u tačnosti iznosi: 0.7% na trening skupu i 0.9% na test skupu. Pored bolje tačnosti, uočava se i bolja sposobnost prepoznavanja MERS virusa (koji biva znatno ređe pogrešno klasifikovan kao SARS2). U slučaju klasifikacije po tipu proteina razlike u tačnosti na trening i test skupu iznose, redom: 0.6% i 2.2%. Razlika je naročito vidljiva kod manje zastupljenih proteina. Model zasnovan na Hamingovom rastojanju je imao problema sa preciznošću pri prepoznavanju proteina ORF7b i ORF10 (koja i na trening i na test skupu pada ispod 20%), dok je model zasnovan na p-adičnoj metrici imao preciznost 100% izuzev u slučaju prepoznavanja ORF10 proteina na trening skupu. U slučaju klasifikacije SARS2 virusa po Svetskoj zdravstvenoj organizaciji nema značajnijih razlika u kvalitetu modela zasnovanih na p-adičnoj metrići, odnosno Hamingovom rastojanju. Na trening skupu model zasnovan na Hamingovom rastojanju ima bolju tačnost 0.5%, ali je razlika na test skupu 0.5% u korist modela zasnovanog na p-adičnoj metrići (test skup je ipak značajniji, jer je reč o podacima koji nisu poznati modelu).

Poređenjem modela sa [fiksnim odsecanjem](#), uočava se značajan pad tačnosti u odnosu na modele sa adaptivnim odsecanjem. Takođe, razlike između modela zasnovanih na p-adičnom i Hamingovom rastojanju nisu značajne.

Dakle, p-adična metrika pozitivno utiče na razlike genetskog kod virusa. Te razlike se najznačajnije ispoljavaju u preciznosti identifikacije manje zastupljenih proteina, ali i sposobnosti razlikovanja MERS soja od SARS2.

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#>
- [6] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html
- [7]https://scikit-learn.org/stable/modules/generated/sklearn.metrics.make_scorer.html
- [8]https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
- [9]<https://scikit-learn.org/stable/modules/svm.html>