

The background of the slide is a dark gray field filled with a complex network of thin, light brown lines. These lines connect numerous small, dark brown circular nodes, creating a web-like or molecular structure. The nodes are distributed across the entire frame, with some appearing more densely connected than others. The overall effect is one of a sophisticated, interconnected system.

Octave i AI funkcionalnosti

Prof. dr Nikola Žarić

Zašto Octave za AI?

Octave je besplatan, open-source alat, veoma sličan MATLAB-u, pogodan za numeričke proračune i rad sa matricama.

Idealno okruženje za brzu izradu prototipa i testiranje AI/ML algoritama.

Posebno pogodan za edukaciju i vizualizaciju podataka.

Nije potrebna instalacija dodatnih biblioteka za osnovne operacije nad podacima.

Prednosti Octave-a u pripremi podataka



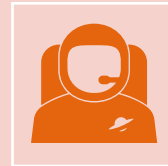
Rad sa matricama i vektorima: Sve operacije su optimizovane za rad sa velikim skupovima podataka.



Jednostavna sintaksa: Brzo kreiranje, modifikacija i vizualizacija podataka.



Ugrađene funkcije: Normalizacija, skaliranje, filtriranje, vizualizacija (scatter, plot, histogram, boxplot).



Brza generacija sintetičkih skupova podataka: Idealno za testiranje i demonstracije algoritama.

Priprema i vizualizacija podataka (primjer)

```
% Generisanje sintetičkog skupa podataka
A = rand(50, 2);          % 50 primjera klase 0
B = rand(50, 2) + 2;      % 50 primjera klase 1 (pomjereni)
C = [A; B];               % Kombinovani skup
labels = [zeros(50,1); ones(50,1)];

% Vizualizacija
scatter(C(:,1), C(:,2), 50, labels, 'filled');
title('Sintetički skup podataka za klasifikaciju');
xlabel('Karakteristika 1');
ylabel('Karakteristika 2');
```

Normalizacija i obrada podataka

- Normalizacija je ključna za mnoge AI algoritme, a u Octave-u se može uraditi u jednoj liniji koda.

```
% Normalizacija podataka (Min-Max)
```

```
C_norm = (C - min(C)) ./ (max(C) - min(C));
```

```
% Prikaz normalizovanih podataka
```

```
scatter(C_norm(:,1), C_norm(:,2), 50, labels, 'filled');
```

```
title('Normalizovani skup podataka');
```

```
xlabel('Karakteristika 1');
```

```
ylabel('Karakteristika 2');
```

Implementacija jednostavne logističke regresije

% Priprema podataka

X = [ones(size(C_norm,1),1) C_norm];

Y = labels;

% Inicijalizacija

theta = zeros(size(X,2),1);

sigmoid = @(z) 1 ./ (1 + exp(-z));

alpha = 0.1;

num_iters = 1000;

Implementacija jednostavne logističke regresije

```
% Gradientni spust
```

```
for i = 1:num_iters
```

```
    h = sigmoid(X*theta);
```

```
    gradient = (1/length(Y)) * (X' * (h - Y));
```

```
    theta = theta - alpha * gradient;
```

```
end
```

```
% Predikcija i prikaz rezultata
```

```
predictions = sigmoid(X*theta) >= 0.5;
```

```
fprintf('Broj tačnih predikcija: %d od %d\n', sum(predictions == Y), length(Y));
```

Vizualizacija granice odlučivanja

```
% Vizualizacija granice odlučivanja  
hold on;  
scatter(C_norm(:,1), C_norm(:,2), 50, labels, 'filled');  
x_vals = linspace(min(C_norm(:,1)), max(C_norm(:,1)), 100);  
y_vals = -(theta(1) + theta(2)*x_vals) / theta(3);  
plot(x_vals, y_vals, 'k-');  
title('Granica odlučivanja - Logistička regresija');  
hold off;
```


Zaključak – Zašto koristiti Octave za pripremu podataka u AI

- Brza i jednostavna priprema i vizualizacija podataka.
- Nema potrebe za dodatnim bibliotekama za osnovne AI zadatke.
- Pogodan za edukaciju, prototipisanje i testiranje algoritama.
- Laka integracija sa drugim alatima i jezicima ako je potrebno

Rukovanje i popunjavanje nedostajućih podataka u Octave-u

Zašto su nedostajući podaci važni?

- Nedostajući podaci su čest problem u realnim skupovima podataka.
- Ako se ignorišu, mogu narušiti preciznost modela ili dovesti do pogrešnih zaključaka.

Kako Octave prepoznaje nedostajuće podatke?

- Nedostajući podaci se označavaju kao NA ili NaN (Not a Number).
- Mogu se pronaći pomoću funkcija `isna` ili `isnan`:

```
x = [13, Inf, NA, NaN];  
isna(x)    % => [0 0 1 0]  
isnan(x)   % => [0 0 0 1]
```

Tehnike popunjavanja nedostajućih podataka:

- Zamjena aritmetičkom sredinom (mean imputation):

```
x = [1 2 NA 4 5];
```

```
x(isna(x)) = mean(x(~isna(x)));
```

- Zamjena medijanom:

```
x(isna(x)) = median(x(~isna(x)));
```

- Zamjena nulom ili drugim fiksnim vrijednostima:

```
x(isna(x)) = 0;
```

Tehnike popunjavanja nedostajućih podataka:

Prednosti Octave-a:

- Jednostavna identifikacija i zamjena nedostajućih vrijednosti.
- Sve operacije nad matricama i vektorima su optimizovane za efikasno rukovanje velikim skupovima podataka.
- Vizualizacija prije i poslije popunjavanja (npr. histogram, scatter plot) pomaže u procjeni uticaja imputacije.
- Naprednije metode (npr. regresija, KNN, Random Forest):
 - Ove metode se češće koriste u Python-u, ali osnovne verzije mogu se implementirati i u Octave-u uz malo više programiranja.

Primjer: Imputacija nedostajućih podataka i vizualizacija

1. Generisanje skupa podataka

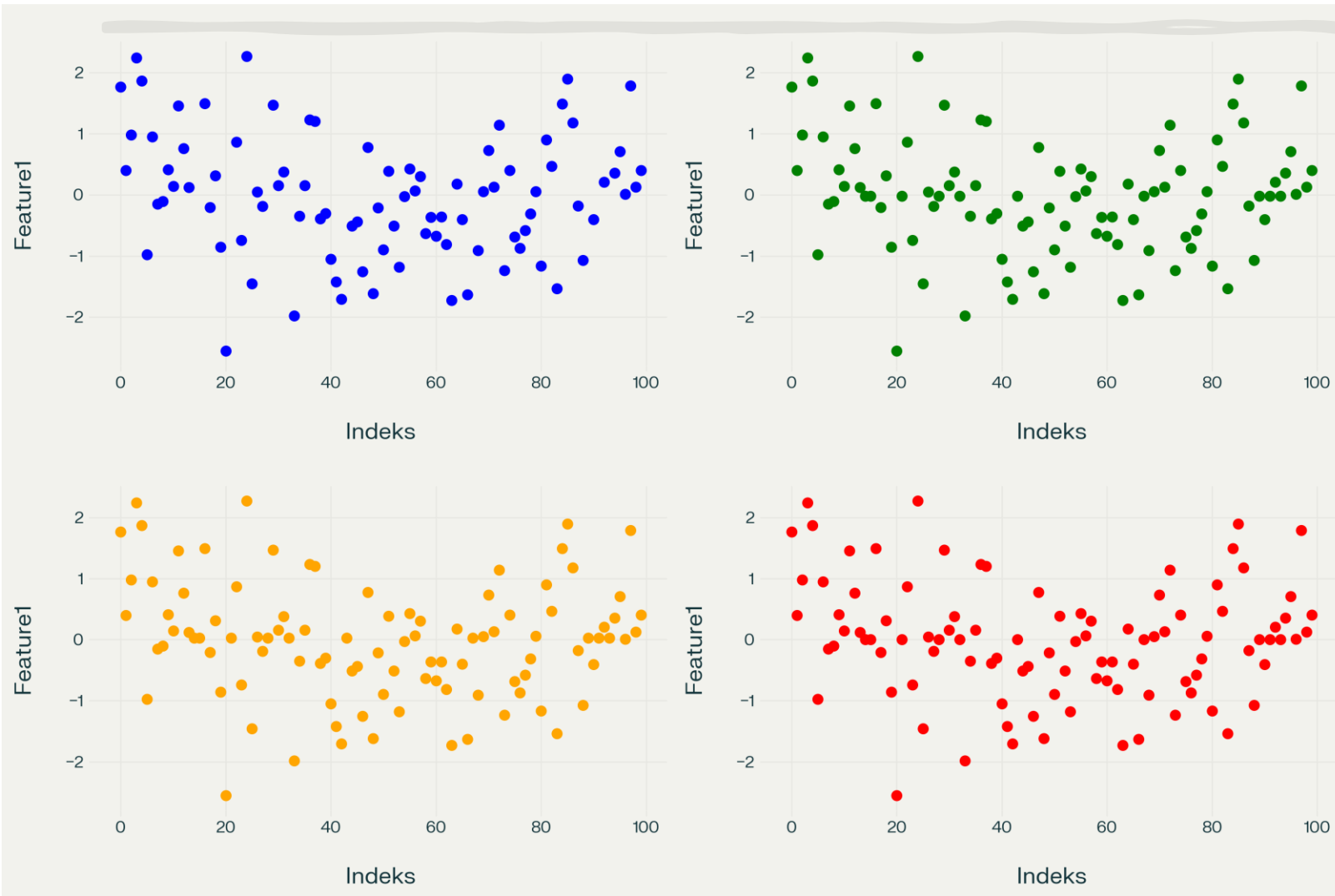
- Kreiran je sintetički skup podataka sa dvije karakteristike (Feature1, Feature2) i 100 primjera. U 10% vrijednosti prve karakteristike (Feature1) namjerno su ubačene nedostajuće vrijednosti (NaN).

2. Tehnike imputacije

Prikazane su tri osnovne tehnike popunjavanja nedostajućih vrijednosti:

- Srednja vrijednost (mean imputation)
- Medijana (median imputation)
- Fiksna vrijednost (npr. 0)

Primjer: Imputacija nedostajućih podataka i vizualizacija



Na slici su prikazana četiri grafa:

- Plavi graf (gore lijevo): Originalni podaci sa NaN vrijednostima
- Zeleni graf (gore desno): Popunjavanje srednjom vrijednošću
- Narandžasti graf (dolje lijevo): Popunjavanje medijanom
- Crveni graf (dolje desno): Popunjavanje sa 0

Primjer: Imputacija nedostajućih podataka i vizualizacija

% Octave primjer: Imputacija nedostajućih podataka i vizualizacija

```
rand('seed', 0); % za reproduktivnost
```

```
data_size = 100;
```

```
X1 = randn(data_size, 1); % normalna distribucija
```

```
X2 = 5 + 2*randn(data_size, 1);
```

% Uvođenje NaN vrijednosti u 10% podataka

```
nan_indices = randperm(data_size, round(0.1*data_size));
```

```
X1(nan_indices) = NaN;
```

% Popunjavanje srednjom vrijednošću

```
mean_val = mean(X1(~isnan(X1)));
```

```
mean_impute = X1;
```

```
mean_impute(isnan(mean_impute)) = mean_val;
```

- % Popunjavanje medijanom

- median_val = median(X1(~isnan(X1)));

- median_impute = X1;

- median_impute(isnan(median_impute)) = median_val;

- % Popunjavanje sa 0

- zero_impute = X1;

- zero_impute(isnan(zero_impute)) = 0;

Primjer: Imputacija nedostajućih podataka i vizualizacija

% Vizualizacija

```
figure;  
subplot(2,2,1);  
scatter(1:data_size, X1, 'b', 'filled');  
title('Originalni podaci sa NaN');  
xlabel('Indeks');  
ylabel('Feature1');  
  
subplot(2,2,2);  
scatter(1:data_size, mean_impute, 'g', 'filled');  
title('Popunjavanje srednjom vrijednošću');  
xlabel('Indeks');  
ylabel('Feature1');
```

```
subplot(2,2,3);  
scatter(1:data_size, median_impute, 'm', 'filled');  
title('Popunjavanje medijanom');  
xlabel('Indeks');  
ylabel('Feature1');  
  
subplot(2,2,4);  
scatter(1:data_size, zero_impute, 'r', 'filled');  
title('Popunjavanje sa 0');  
xlabel('Indeks');  
ylabel('Feature1');  
  
sgtitle('Imputacija nedostajućih podataka – poređenje tehnika');
```