```
================================================================================
====
GENERATES 3D PROJECTILE DATASET
================================================================================
====
load('3D-projectile.mat')

N = 3000

num_launches = 750;

dt = 1/N; %1 second divided by the number of iterations

index = 1;

for k = 1 : num_launches

%initial positions
x0 = 0;
y0 = 0;
z0 = 1.8 + .1*rand();

x(1) = x0;
y(1) = y0;
z(1) = z0;

%x and y velocities

if(rand() >= .75)

vx = 35 + 5*rand();
vy = 30 + 2.5*rand();

else

vx = 28 + 2.5*rand();
vy = 30 + 5*rand();

endif

for i = 2 : N

%z velocity

vz = -9.8*i*dt;

x(i) = x(i-1) + vx*dt;
y(i) = y(i-1) + vy*dt;
```

```
z(i) = z(i-1) + vz*dt;

endfor

temp = [x y z];

data_array1{index} = temp;

index = index + 1;

endfor



========================================
DISPLAYS DATA
========================================


m = 8

X = [];
Y = [];
Z = [];
S = [];
C = [];


for i = 1 : m

temp = data_array1{i};

x(1:N) = temp(1 : N);
y(1:N) = temp(N + 1 : 2*N);
z(1:N) = temp(2*N + 1 : 3*N);

s(1:N) = 15;
c(1:N,1) = rand();
c(1:N,2) = rand();
c(1:N,3) =  rand();

X = [X x];
Y = [Y y];
Z = [Z z];
S = [S s];
C = [C; c];

endfor
```

```
scatter3(X,Y,Z,S,C,'.')
view(160,20)
title('Figure 8')


============================================================
GENERATES DATA TREE
============================================================


dimension = 3*N;

[category_tree delta_tree anchor_tree] = generate_data_tree_N(data_array,dimension);


============================================================
GENERATES NEW ITEM
============================================================


%initial positions
x0 = 0;
y0 = 0;
z0 = 1.8 + .1*rand();

x(1) = x0;
y(1) = y0;
z(1) = z0;

%x and y velocities

if(rand() >= .75)

vx = 35 + 5*rand();
vy = 30 + 2.5*rand();

else

vx = 28 + 2.5*rand();
vy = 30 + 5*rand();

endif

for i = 2 : N

%z velocity
```

```
vz = -9.8*i*dt;

x(i) = x(i-1) + vx*dt;
y(i) = y(i-1) + vy*dt;
z(i) = z(i-1) + vz*dt;

endfor

scatter3(x,y,z, '.')
view(160,20)

new_data_item{1} = [x y z];


=========================================================
GENERATES PREDICTIONS
=========================================================


M = 1750;

missing_data_vector = [M + 1 : N]; %generates missing x data
missing_data_vector = [missing_data_vector (N + 2 + M : 2*N)]; %generates missing y
data
missing_data_vector = [missing_data_vector (2*N + 2 + M : 3*N)]; %generates missing
z data

[category_index predicted_vector final_delta min_difference predicted_vector_array]
= predict_best_fit_tree_N(anchor_tree, delta_tree, new_data_item, 1, 1,
missing_data_vector, dimension);

num_predictions = size(predicted_vector_array,2)


=========================================================
DISPLAYS PREDICTIONS
=========================================================


X = [];
Y = [];
Z = [];
S = [];
C = [];

%loads the best-fit match
```

```
temp = predicted_vector;

x(1:N) = temp(1 : N);
y(1:N) = temp(N + 1 : 2*N);
z(1:N) = temp(2*N + 1 : 3*N);

s(1:N) = 85;
c(1:N,1) = rand();
c(1:N,2) = rand();
c(1:N,3) = rand();

X = x;
Y = y;
Z = z;
S = s;
C = c;

figure

hold

scatter3(X,Y,Z,S,C,'.')

%calculates and loads the average of all matches

avg_vector = predicted_vector_array{1};

for i = 2 : num_predictions

avg_vector = avg_vector + predicted_vector_array{i};

endfor

avg_vector = avg_vector / num_predictions;

temp = avg_vector;

x(1:N) = temp(1 : N);
y(1:N) = temp(N + 1 : 2*N);
z(1:N) = temp(2*N + 1 : 3*N);

s(1:N) = 85;
c(1:N,1) = rand();
c(1:N,2) = rand();
c(1:N,3) = rand();

X = x;
Y = y;
```

```
Z = z;
S = s;
C = c;

scatter3(X,Y,Z,S,C,'.')

%loads the actual path

temp = new_data_item{1};

x(1:N) = temp(1 : N);
y(1:N) = temp(N + 1 : 2*N);
z(1:N) = temp(2*N + 1 : 3*N);

s(1:N) = 85;
c(1:N,1) = rand();
c(1:N,2) = rand();
c(1:N,3) = rand();

X = x;
Y = y;
Z = z;
S = s;
C = c;

scatter3(X,Y,Z,S,C,'.')
legend('best-fit', 'average fit', 'actual path')

view(100,20)



============================================================
CALCULATES AND DISPLAYS ERROR
============================================================


actual_vector = new_data_item{1};

clear avg_vector

cnt = 1;

for M = 0 : 25 : 3000

missing_data_vector = [M + 1 : N]; %generates missing x data
missing_data_vector = [missing_data_vector (N + 2 + M : 2*N)]; %generates missing y
data
```

```
missing_data_vector = [missing_data_vector (2*N + 2 + M : 3*N)]; %generates missing
z data

[category_index predicted_vector final_delta min_difference predicted_vector_array]
= predict_best_fit_tree_N(anchor_tree, delta_tree, new_data_item, 1, 1,
missing_data_vector, dimension);

avg = 0;

for i = M + 1 : 3*N

avg = avg + abs(predicted_vector(i) - actual_vector(i))/(.0001 +
abs(actual_vector(i)));

endfor

avg = avg/(3*N);

avg_vector(cnt) = avg;

cnt = cnt + 1;

endfor


plot(0 : 25 : 3000, 100*avg_vector)
```