function [best\_feature\_val] = optimizeContentRating(fin\_model\_low, fin\_model\_med, fin\_model\_high, crit\_model, movie, objParams)

%Returns the optimal Content Rating

% This will return the optimal content rating, while enforcing the rule

% that only one content rating can actually be selected

%Get the current budget being considered (since this won't change)

%This will tell us which model to use here:

if movie{1, 'AdjustedBudget'} < 11

internal\_fin\_model = fin\_model\_low;

elseif movie{1, 'AdjustedBudget'} >= 11 && movie{1, 'AdjustedBudget'} <= 75

internal\_fin\_model = fin\_model\_med;

else

internal\_fin\_model = fin\_model\_high;

end

obj\_values\_vec = zeros(1, 6);

content\_rating\_set = {'PG\_13', 'PG', 'R', 'G', 'NC\_17', 'NR'};

num = length(content\_rating\_set);

for i=1:num

movie(1, content\_rating\_set{i}) = {0};

end

movie(1, content\_rating\_set{1}) = {1};

box\_office = predict(internal\_fin\_model, movie);

critic = predict(crit\_model, movie);

obj\_value = getObjective(box\_office, critic, objParams);

obj\_values\_vec(1) = obj\_value;

for i=2:num

movie(1, content\_rating\_set{i-1}) = {0};

movie(1, content\_rating\_set{i}) = {1};

box\_office = predict(internal\_fin\_model, movie);

critic = predict(crit\_model, movie);

obj\_value = getObjective(box\_office, critic, objParams);

obj\_values\_vec(i) = obj\_value;

end

highest\_index = find(obj\_values\_vec==max(obj\_values\_vec));

best\_feature\_val = content\_rating\_set{highest\_index};

% %Find objective for each of these in order:

% %For PG\_13:

% movie(1, 'PG\_13') = {1};

% movie(1, 'PG') = {0};

% movie(1, 'R') = {0};

% movie(1, 'G') = {0};

% movie(1, 'NC\_17') = {0};

% movie(1, 'NR') = {0};

% %Run the random forest model for box office:

% box\_office = predict(internal\_fin\_model, movie);

% %Run the random forest model for metacritic:

% critic = predict(crit\_model, movie);

% %Get the objective function value:

% obj\_value = getObjective(box\_office, critic, objParams);

% %Store this objective value:

% obj\_values\_vec(1) = obj\_value;

%

% %For PG:

% movie(1, 'PG\_13') = {0};

% movie(1, 'PG') = {1};

% %Run the random forest model for box office:

% box\_office = predict(internal\_fin\_model, movie);

% %Run the random forest model for metacritic:

% critic = predict(crit\_model, movie);

% %Get the objective function value:

% obj\_value = getObjective(box\_office, critic, objParams);

% %Store this objective value:

% obj\_values\_vec(2) = obj\_value;

%

% %For R:

% movie(1, 'PG') = {0};

% movie(1, 'R') = {1};

% %Run the random forest model for box office:

% box\_office = predict(internal\_fin\_model, movie);

% %Run the random forest model for metacritic:

% critic = predict(crit\_model, movie);

% %Get the objective function value:

% obj\_value = getObjective(box\_office, critic, objParams);

% %Store this objective value:

% obj\_values\_vec(3) = obj\_value;

%

% %For G:

% movie(1, 'R') = {0};

% movie(1, 'G') = {1};

% %Run the random forest model for box office:

% box\_office = predict(internal\_fin\_model, movie);

% %Run the random forest model for metacritic:

% critic = predict(crit\_model, movie);

% %Get the objective function value:

% obj\_value = getObjective(box\_office, critic, objParams);

% %Store this objective value:

% obj\_values\_vec(4) = obj\_value;

%

% %For NC\_17:

% movie(1, 'G') = {0};

% movie(1, 'NC\_17') = {1};

% %Run the random forest model for box office:

% box\_office = predict(internal\_fin\_model, movie);

% %Run the random forest model for metacritic:

% critic = predict(crit\_model, movie);

% %Get the objective function value:

% obj\_value = getObjective(box\_office, critic, objParams);

% %Store this objective value:

% obj\_values\_vec(5) = obj\_value;

%

% %For NR:

% movie(1, 'NC\_17') = {0};

% movie(1, 'NR') = {1};

% %Run the random forest model for box office:

% box\_office = predict(internal\_fin\_model, movie);

% %Run the random forest model for metacritic:

% critic = predict(crit\_model, movie);

% %Get the objective function value:

% obj\_value = getObjective(box\_office, critic, objParams);

% %Store this objective value:

% obj\_values\_vec(6) = obj\_value;

%

% %Retrieve the index that yielded the highest objective value:

% highest\_index = find(obj\_values\_vec==max(obj\_values\_vec))

% %Return the feature value that corresponds to the best one...

% if highest\_index == 1

% best\_feature\_val = 'PG\_13';

% elseif highest\_index == 2

% best\_feature\_val = 'PG';

% elseif highest\_index == 3

% best\_feature\_val = 'R';

% elseif highest\_index == 4

% best\_feature\_val = 'G';

% elseif highest\_index == 5

% best\_feature\_val = 'NC\_17';

% elseif highest\_index == 6

% best\_feature\_val = 'NR';

% end

end