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

%Returns the optimal genre

% This will return the optimal genre, while enforcing the rule that only one content rating can actually be selected

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, 15);

genre\_set = {'Adventure', 'Action', 'Animation', 'Family',...

'ScienceFiction', 'Comedy', 'Drama', 'Romance', 'Horror', 'Mystery',...

'Documentary', 'Music', 'Thriller', 'Crime', 'War'};

num = length(genre\_set);

%First, zero out evry genre:

for i=1:num

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

end

%Now, calcualte the objective values

%For the first time (just Adventure):

movie(1, genre\_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;

%Next, calculate the objectivs for the remainding genres:

for i=2:num

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

movie(1, genre\_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

%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:

best\_feature\_val = genre\_set{highest\_index};

end