%Purpose: Optimization model for the project

%This model will assume no/little interaction between the features...

%Fix one of them, keep incrementing until its optimum is reached, and then

%move on the next feature to do the same

%This is going to be done in parallel loops (additive)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%% DECLARE PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Random Forest Parameters

numTrees = 50;

%CSV File Path Parameters

LowBudgetCSV = 'MovieData\_LowBudget.csv';

MediumBudgetCSV = 'MovieData\_MediumBudget.csv';

HighBudgetCSV = 'MovieData\_HighBudget.csv';

AllCSV = 'MovieDataAll.csv';

movieConstructionCSV = 'MovieConstruction.csv');

%Objective Function Parameters

weightBoxOffice = 0.5;

weightMetacritic = 0.5;

sigmaBoxOffice = 1;

sigmaMetacritic = 1;

meanBoxOffice = 1;

meanMetacritic = 1;

%Optimization Model Constraint Parameters

%Budget

budget\_UB = 100; %USER CHOSEN

budget\_LB = 0;

budget\_Step = 2;

%Theatres (widest release)

theatres\_UB = 1000; %USER CHOSEN

theatres\_LB = 0;

theatres\_Step = 5;

%Franchise (1 for must be franchise; 0 for must NOT be; -1 for anything)

franchise = 1; %USER CHOSEN

%Google trends

trends\_UB = 50; %USER CHOSEN

trends\_LB = 0;

trends\_Step = 2;

%Sentiment

Sentiment\_UB = 1;

Sentiment\_LB = -1;

sentiment\_Step = 0.1;

%Number of nominated actors

numNominated\_UB = 5; %USER CHOSEN

numNominated\_LB = 0;

numNominated\_Step = 1;

%Number of winning actors

numWinning\_UB = 5; %USER CHOSEN

numWinning\_LB = 0;

numWinning\_Step = 1;

%Total number of nominations across cast

totalNominated\_UB = 20;

totalNominated\_LB = 0;

totalNominated\_Step = 2;

%Total number of winning

totalWinning\_UB = 10;

totalWinning\_LB = 0;

totalWinning\_Step = 2;

%Months (USER CHOSEN) --> 1 if allowed, 0 else

jan = 1;

feb = 1;

mar = 1;

apr = 1;

may = 1;

jun = 1;

jul = 1;

aug = 1;

sep = 1;

oct = 1;

nov = 1;

dec = 1;

%Runtime

runtime\_UB = 300; %USER CHOSEN

runtime\_LB = 0; %USER CHOSEN

%Genres (USER CHOSEN) --> 1 if allowed, 0 else

adventure = 1;

action = 1;

animation = 1;

family = 1;

scifi = 1;

comedy = 1;

drama = 1;

romance = 1;

horror = 1;

mystery = 1;

documentary = 1;

music = 1;

thriller = 1;

crime = 1;

war = 1;

%Total Number of Languages

languages\_UB = 10; %USER CHOSEN

languages\_LB = 1;

languages\_Step = 1;

%Content Rating (USER CHOSEN) --> 1 if allowed, 0 else

PG13 = 1;

PG = 1;

R = 1;

G = 1;

NC17 = 1;

NR = 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%% READ TABLES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Read all of the tables

%Framework for the multiple BOX OFFICE MODELS:

%Read in low budget data set

movieDataLow = readtable(LowBudgetCSV);

%Block out title and targets

movieDataLow.AdjustedWorldwide = [];

movieDataLow.metacritics=[];

movieDataLow.title=[];

%Read in medium budget data set

movieDataMedium = readtable(MediumBudgetCSV);

%Block out the titles and targets

movieDataMedium.AdjustedWorldwide = [];

movieDataMedium.metacritics=[];

movieDataMedium.title=[];

%Read in high budget data set

movieDataHigh = readtable(HighBudgetCSV);

%Block out the titles and targets:

movieDataHigh.AdjustedWorldwide = [];

movieDataHigh.metacritics=[];

movieDataHigh.title=[];

%Framwork for the metacritic model(s):

movieDataAll = readtable(AllCSV);

movieDataAll.AdjustedWorldwide = [];

movieDataAll.metacritics=[];

movieDataAll.title=[];

%Framework for the movie construction vector:

movieConstruction = readtable(movieConstructionCSV);

movieConstruction.AdjustedWorldwide = [];

movieConstruction.metacritics=[];

movieConstruction.title=[];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%% TRAIN MODELS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Note: No need to split into test and train because we're using all data

%Train a model with all of the data:

%TRAIN LOW BUDGET MODEL

modelLow = TreeBagger(numTrees, movieDataLow, 'AdjustedDomestic','Method','regression');

%TRAIN MEDIUM BUDGET MODEL

modelMedium = TreeBagger(numTrees, movieDataMedium, 'AdjustedDomestic','Method','regression');

%TRAIN HIGH BUDGET MODEL

modelHigh = TreeBagger(numTrees, movieDataHigh, 'AdjustedDomestic','Method','regression');

%TRAIN METACRITIC MODEL (using all data)

modelAll = TreeBagger(numTrees, movieDataAll, 'AdjustedDomestic','Method','regression');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%% OPTIMIZATION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Pick starting vector as the LB of all variables

movieConstruction(1, 'Adjusted Budget') = {budget\_LB};

movieConstruction(1, 'Num\_Theatres\_Widest') = {theatres\_LB};

if franchise == -1:

movieConstruction(1, 'Franchise\_Raw') = {0};

else

movieConstruction(1, 'Franchise\_Raw') = {franchise};

end

movieConstruction(1, 'Google\_Trends') = {trends\_LB};

movieConstruction(1, 'sentiment') = {sentiment\_LB};

movieConstruction(1, 'numNominatedActors') = {numNominated\_LB};

movieConstruction(1, 'numWinningActors') = {numWinning\_LB};

movieConstruction(1, 'totNumNom') = {totalNominated\_LB};

movieConstruction(1, 'totNumWins') = {totalWinning\_LB};

movieConstruction(1, 'Mon') = {0};

movieConstruction(1, 'Tue') = {0};

movieConstruction(1, 'Wed') = {0};

movieConstruction(1, 'Thurs') = {0};

movieConstruction(1, 'Fri') = {0};

movieConstruction(1, 'Sat') = {0};

movieConstruction(1, 'Sun') = {0};

movieConstruction(1, 'Jan') = {0};

movieConstruction(1, 'Feb') = {0};

movieConstruction(1, 'Mar') = {0};

movieConstruction(1, 'Apr') = {0};

movieConstruction(1, 'May') = {0};

movieConstruction(1, 'Jun') = {0};

movieConstruction(1, 'Jul') = {0};

movieConstruction(1, 'Aug') = {0};

movieConstruction(1, 'Sep') = {0};

movieConstruction(1, 'Oct') = {0};

movieConstruction(1, 'Nov') = {0};

movieConstruction(1, 'Dec') = {0};

movieConstruction(1, 'runtime') = {runtime\_LB};

movieConstruction(1, 'Adventure') = {adventure};

movieConstruction(1, 'Action') = {action};

movieConstruction(1, 'Animation') = {animation};

movieConstruction(1, 'Family') = {family};

movieConstruction(1, 'Science Fiction') = {scifi};

movieConstruction(1, 'Comedy') = {comedy};

movieConstruction(1, 'Drama') = {drama};

movieConstruction(1, 'Romance') = {romance};

movieConstruction(1, 'Horror') = {horror};

movieConstruction(1, 'Mystery') = {mystery};

movieConstruction(1, 'Documentary') = {documentary};

movieConstruction(1, 'Music') = {music};

movieConstruction(1, 'Thriller') = {thriller};

movieConstruction(1, 'Crime') = {crime};

movieConstruction(1, 'War') = {war};

movieConstruction(1, 'total\_language') = {languages\_LB};

%movieConstruction(1, 'total\_countries') = {0};

movieConstruction(1, 'total\_language') = {languages\_LB};

movieConstruction(1, 'PG-13') = {PG13};

movieConstruction(1, 'PG') = {PG};

movieConstruction(1, 'R') = {R};

movieConstruction(1, 'G') = {G};

movieConstruction(1, 'NC17') = {NC17};

movieConstruction(1, 'NR') = {NR};

%NEXT STEP:

%Make a function that iterates and finds the best value

%Feed it LB, UB, step size, etc.

%Pick a starting movie vector

%Compute the objective:

% First, compute the metacritic score (only one model)

% Then, conditional check on budget and pipeline split

% Use the appropriate model for box office

% call the getObjective function to get the objective value

%Fix all except the first feature

%For all iterations in that feature:

% Increment that feature from min to max value for that feature

% On each iteration:

% Record the feature value that gives the highest