Ava Exelbirt and Samuel Reade ~ Statistical Learning Project 2

Motivation and Goal: As the entertainment industry grows, predicting a movie's success is vital for content platforms and production companies. Success is typically measured by revenue, budget, and audience reception. Accurate prediction is challenging due to factors like budget, genre, release time, and audience engagement. By analyzing historical data, teams can identify patterns that guide investments, marketing, and content strategies. This project aims to develop classification models, including Logistic Regression, Random Forest, Decision Trees, Gradient Boosting, Neural Networks, LDA, QDA, and Naive Bayes, to predict movie success. It also focuses on feature selection and model interpretation using methods like SHAP, Recursive Feature Elimination, cross-validation, and partial dependence plots. Key features analyzed include budget, release year, runtime, and popularity. The ultimate goal is to create a high-accuracy model that helps stakeholders make data-driven decisions and understand the factors behind movie success.

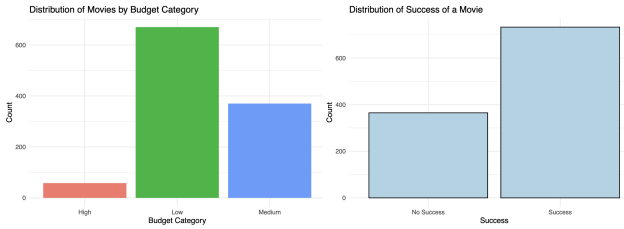
Our Process: We began by scraping the web for an interesting dataset and defining our research goals. After selecting the Horror dataset, we cleaned it by removing outliers and NA values. Feature engineering included creating boolean columns, converting categorical variables to factors, and defining our target variable, **success**, based on whether profit (revenue - budget) was greater than zero. We analyzed correlations, conducted exploratory data analysis using visualizations, and split the data into training (60%) and testing (40%) sets. A normalized dataset was also prepared for specific classification methods. Models were fitted, predictions made, and evaluations performed, including PDP plots. Feature selection and predictor comparisons were conducted using various techniques. Collaboration was managed through a shared GitHub repository. Each person worked on sections, committing and pushing changes, while the other pulled updates to continue work. Weekly meetings were held to review progress, provide feedback, and make edits.

About the Dataset: The dataset, scraped from TidyTuesday, contains information on ~32,540 horror movies dating back to the 1950s, sourced from "The Movie Database" via the tmdb API. It includes 20 variables detailing movie performance, reception, and themes. These features will train a classification model to predict movie success, focusing on identifying key predictors for accurate classification.

Data Dictionary: The features used in classification

1. The **release_date** variable is a date field that records the date when the movie was first released.
2. The **popularity** variable is a numerical value representing the movie's popularity score based on audience interactions.
3. The **budget** variable is an integer field capturing the movie's production budget in USD.
4. The **revenue** variable is an integer field indicating the total revenue earned by the movie in USD.
5. The **runtime** variable is an integer field that specifies the duration of the movie in minutes.

Data Preprocessing Step: During preprocessing, we addressed missing values and removed irrelevant columns. NA values were handled by dropping unnecessary columns and replacing numeric NAs with the median to preserve distribution while minimizing outlier influence. For character columns, missing values were replaced with "Unknown" to retain data without bias. We removed columns like id, poster_path, backdrop_path, and collection_name, as they were identifiers or redundant. These steps ensured the dataset was clean and ready for analysis, retaining enough observations for robust results.



Feature Engineering: We performed feature engineering to enhance the dataset's usability. Boolean-like columns, such as *adult*, were converted to logical types, and categorical variables like *original_language* and *status* were transformed into factors. A new variable, *release_year*, was extracted from *release_date* for temporal analysis. Outliers in *runtime* were replaced with 0 if outside the IQR, and *popularity* values exceeding 10,000 were removed as outliers. The *budget* column was categorized into Low, Medium, and High for easier comparisons. We engineered the target variable by calculating *profit* (revenue - budget) and creating a binary *success* variable: movies with positive profit were labeled "Success," and others "No Success." Most movies (700) were classified as Success, with 375 as No Success. Around 650 movies had low budgets, 375 medium, and only 50 high. These features form the foundation of our predictive modeling.

```
original_title       title            original_language   overview
Length:659         Length:659         en     :560        Length:659
Class :character   Class :character   ja     : 16        Class :character
Mode  :character   Mode  :character   es     : 14        Mode  :character
                                      hi     : 14
                                      ko     : 10
                                      de     :  8
                                      (Other): 37
  tagline           release_date          popularity        vote_count
Length:659         Min.   :1953-06-05   Min.   :   0.600   Min.   :   0.0
Class :character   1st Qu.:1994-12-12   1st Qu.:   7.186   1st Qu.:  98.5
Mode  :character   Median :2007-01-19   Median :  15.516   Median :  543.0
                   Mean   :2003-08-09   Mean   :  27.011   Mean   : 1206.6
                   3rd Qu.:2015-01-28   3rd Qu.:  31.030   3rd Qu.: 1544.5
                   Max.   :2022-09-29   Max.   :1071.398   Max.   :16900.0

 vote_average       budget             revenue             runtime
Min.   : 0.000   Min.   :       1   Min.   :       1   Min.   :  0.00
1st Qu.: 5.300   1st Qu.:  1000000   1st Qu.:  675326   1st Qu.: 87.50
Median : 6.000   Median :  5000000   Median : 11642254   Median : 95.00
Mean   : 5.797   Mean   : 12769148   Mean   : 39044327   Mean   : 91.09
3rd Qu.: 6.600   3rd Qu.: 15000000   3rd Qu.: 45023606   3rd Qu.:103.00
Max.   :10.000   Max.   :200000000   Max.   :701842551   Max.   :153.00

          status      adult                          genre_names
In Production  :  0   Mode :logical   Horror, Thriller         :103
Planned        :  0   FALSE:659       Horror                   : 99
Post Production:  0                   Horror, Mystery, Thriller: 52
```

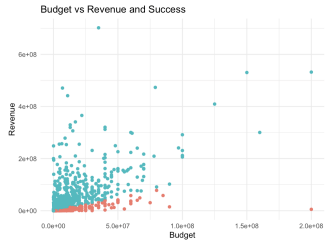
```
Released       :659

 release_year  budget_category
Min.   :1953   Length:659
1st Qu.:1994   Class :character
Median :2007   Mode  :character
Mean   :2003
3rd Qu.:2015
Max.   :2022

                        Comedy, Horror           : 36
                        Horror, Science Fiction  : 29
                        Drama, Horror, Thriller  : 27
                        (Other)                  :313
      profit                    success
Min.   :-194775779   No Success:212
1st Qu.:    -99162   Success   :447
Median :   3400000
Mean   :  26275179
3rd Qu.:  30288153
Max.   : 666842551
```

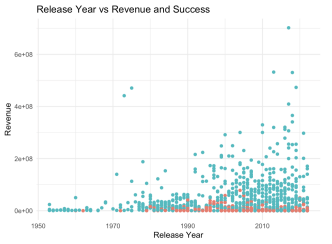
```
No Success   Success
 0.3216995   0.6783005
```

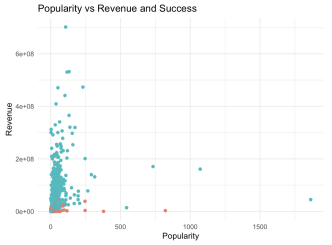
Split the Data: We will split the data into training (60%) and testing (40%) sets. We will then look at the new data by checking the number of rows in training and testing sets and looking at the summary of the training set. We will use the training set to train our classification models and the testing set to evaluate our models on new data. We also split and trained the data that was later normalized for the methods that needed it. 67.83% of the training data is successful movies, 32.17% of the training data are unsuccessful movies: semi unbalanced dataset.

Normalization: We chose normalization over standardization due to the limited data and skewed distributions. Min-max normalization was applied to the budget, release_year, runtime, and popularity columns df_for_class, scaling them to a range between 0 and 1. This ensures all numeric features are on the same scale, preventing variables with larger magnitudes from dominating in machine learning models. We used the normalized datasets for training and testing across QDA, LDA, logistic regression, multinomial logistic regression, and neural network models. **EDA:** We performed exploratory data analysis, visualizing the distributions of numeric and categorical features, target variables, and relationships between predictors and

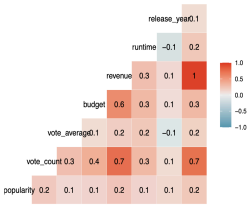
response variables. A positive linear relationship was observed between revenue and budget, with most observations classified as "no success" at lower revenue and budget levels. There was a weaker correlation between release year and success, with no clear pattern of success across years. A strong linear relationship between revenue and popularity was seen, with no successes at low popularity and revenue. No strong correlation was found between runtime and success, as successes were evenly distributed across runtimes with low revenues.







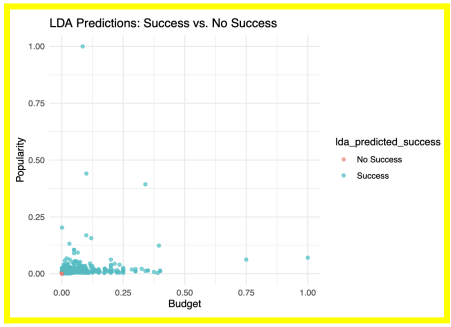
Correlation plot: The correlation plot shows pairwise correlation coefficients between numeric variables in the horror dataset, highlighting how features relate to each other. Notably, there is a perfect positive correlation between profit and revenue, with both increasing proportionally. Strong positive correlations are also seen between profit and vote count, and revenue and vote count. Additionally, there is a negative correlation between release year and both runtime and vote average.



Classification with Emphasis on Prediction:

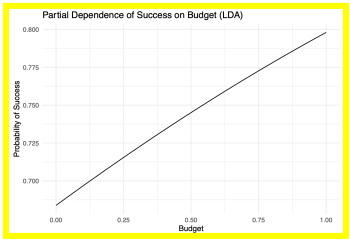
Classifying Success with LDA:

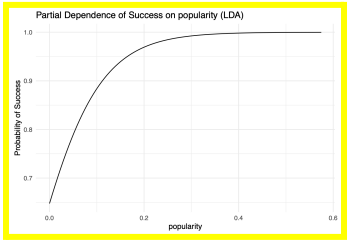
```
## Call:
## lda(success ~ budget + release_year + runtime + popularity, data = training_n)
##
## Prior probabilities of groups:
## No Success    Success
##  0.3216995  0.6783005
##
## Group means:
##                budget release_year   runtime  popularity
## No Success 0.05370571     0.7686628 0.4738326 0.006290148
## Success    0.06865487     0.7056058 0.5254771 0.017916457
##
## Coefficients of linear discriminants:
##                    LD1
## budget        1.005574
## release_year -2.732860
## runtime       2.633095
## popularity   23.712963
```

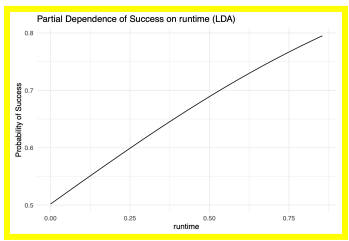


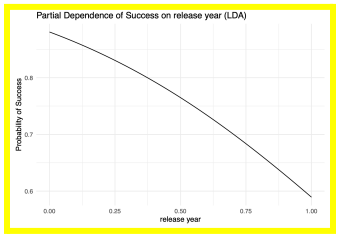
	Actual	
Predicted	No Success	Success
No Success	19	16
Success	134	270

The accuracy of this model is about 65.83%. This performs moderately well on new data.





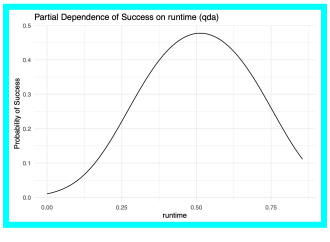


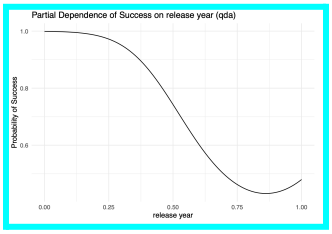


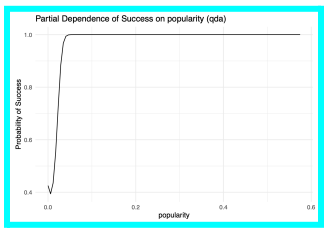
The LDA model used budget, release year, runtime, and popularity to classify success. Budget and runtime had a small positive influence on success with a loading of about 1 and 2.6 respectively. Release year had a negative influence on success with a coefficient of -2.7. The popularity had a very large influence on success with a 23.7 coefficient. The PDP plots reflect these findings showing the influence each variable had on the classification of success. ***Classifying Success with QDA:*** The accuracy of this model is about 62.41%. This performs moderately well on new data. The output of the QDA model shows the average value of each feature within each class. For budget, runtime, and popularity, higher values correlate to success. It also shows that movies released more recently are less likely to be successful. However, for runtime it seems like there is a golden area for the length of movies. Budget's PDP plot reflects that lower values indicate more success which contradicts the QDA output surprisingly. For popularity and release year the PDP plots seem accurate.

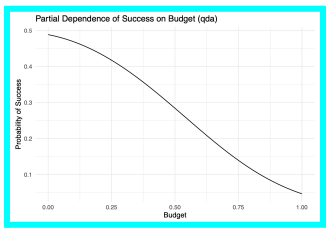
	Actual	
Predicted	No Success	Success
No Success	117	129
Success	36	157

```
## Prior probabilities of groups:
## No Success    Success
##  0.3216995  0.6783005
##
## Group means:
##                budget release_year   runtime  popularity
## No Success 0.05370571     0.7686628 0.4738326 0.006290148
## Success    0.06865487     0.7056058 0.5254771 0.017916457
```







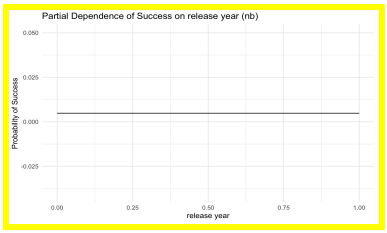


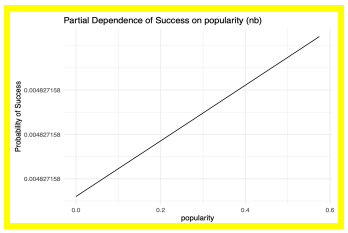
Classifying Success with Naive Bayes: Naive Bayes is a probabilistic model based on Bayes' Theorem. The model assumes that the features are conditionally independent given the target variable, success. The accuracy of this model is about 63.10%. This performs moderately well on new data. Similar to LDA and QDA the output shows each variable described by the mean and variance. Naive Bayes calculates the probability of each observation belonging to each class. Similar to LDA and QDA budget, runtime, and popularity have positive correlations to success while release year has a neutral correlation.

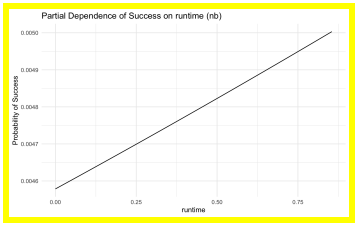
```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
No Success    Success
 0.3216995  0.6783005

Conditional probabilities:
            budget
Y                [,1]     [,2]
  No Success 10741143 19918339
  Success    13730975 19930261

            release_year
Y                [,1]     [,2]
  No Success 2006.038 11.47751
  Success    2001.687 15.63314
```

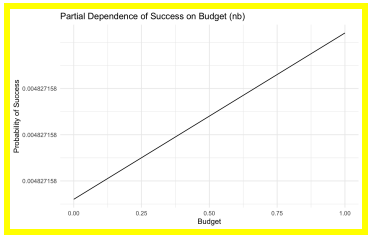
```
            runtime
Y                [,1]     [,2]
  No Success 84.81604 32.49614
  Success    94.06040 22.54887

            popularity
Y                [,1]     [,2]
  No Success 12.31872 20.38792
  Success    33.97886 64.34544
```

	Actual	
Predicted_NB	No Success	Success
No Success	107	116
Success	46	170









Classifying Success with Shrinkage: Lasso helps with feature selection and Ridge Regression helps with handling multicollinearity. ***Ridge Regression:*** Ridge regression reduces variance in the presence of correlated predictors like budget and popularity by shrinking their coefficients. This reflects each predictor's importance, with popularity having the highest coefficient, indicating its stronger influence on success. Filmmakers benefit from considering all factors, even those with smaller effects. We'll use cross-validation to find the optimal lambda and then train the model with it. We found that the optimal lambda for Ridge regression is about 0.00858. The accuracy of this model is about 69.35%. This performs moderately well on new data. We can see that popularity has a coefficient of about 53.72, while the other variables have coefficients closer to 0. This shows that popularity has the biggest (positive) influence on predicting success.

	Actual	
Predicted	0	1
0	40	30
1	172	417

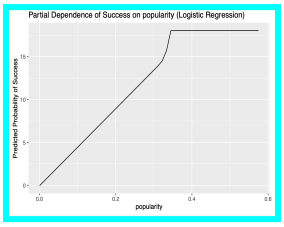
	s0
(Intercept)	1.1558322
budget	-0.3046535
release_year	-1.7238692
runtime	0.6027949
popularity	53.7195835

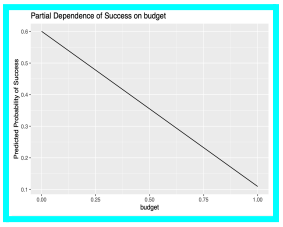
	s0
(Intercept)	1.34153592
budget	-0.89243201
release_year	-1.94555545
runtime	0.06609342
popularity	87.50579005

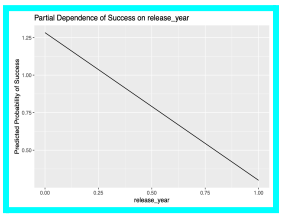
Lasso Regression: We will fit the Lasso model using cross-validation to find the optimal lambda, then train the model with it. The optimal lambda for Ridge regression is 0.00068, yielding an accuracy of 68.44%, which performs moderately well on new data. Lasso regression automatically eliminates less important predictors like budget, making the model more interpretable. It confirms Ridge's finding that popularity has the largest coefficient (87.51), showing a strong positive relationship with success. This insight helps filmmakers focus on key factors, reducing unnecessary expenses.

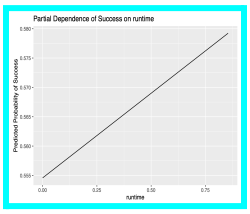
	Actual	
Predicted	0	1
0	60	56
1	152	391

Classifying Success with Logistic Regression: The accuracy of this model is about 65.15%. This performs moderately well on new data. Once again popularity has a high correlation with success, rising fast then flattening. Runtime also is highly correlated with success. Budget's PDP shows a negative correlation which might be because popularity has such a high correlation coefficient. Release year is negative again as well. The output of the logistic regression models illustrates what the PDP plots show. The runtime and popularity have positive influences on success while budget and release year have negative effects. We believe popularity has corrupted this model into only using that feature to classify.










```
Call:  glm(formula = success ~ budget + release_year + runtime + popularity,
    family = "binomial", data = training_lg)

Coefficients:
 (Intercept)        budget  release_year       runtime    popularity
     1.35118      -0.98439      -1.97016       0.05776      89.83537

Degrees of Freedom: 658 Total (i.e. Null);  654 Residual
Null Deviance:      827.9
Residual Deviance: 720.3        AIC: 730.3
```

	Actual	
Predicted	0	1
0	39	39
1	114	247

Classifying Budget into four predicted groups with multinomial logistic regression:

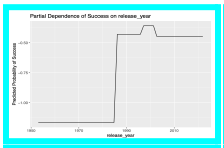
```
Call:
multinom(formula = budget_category ~ revenue + release_year +
    popularity, data = training_n)

Coefficients:
          (Intercept)      revenue release_year  popularity
Medium     -0.2282491 3.319914e-08   -0.3397700 0.003404545
High       -0.6738019 5.083066e-08   -0.2738802 0.014257640
Very High  -1.1782284 6.173187e-08   -0.5745488 0.026366845

Std. Errors:
           (Intercept)      revenue release_year   populari
Medium    2.204654e-16 6.207521e-09 1.501721e-16 3.472101e-
High      2.082177e-16 5.953165e-09 1.438903e-16 3.366093e-
Very High 1.549093e-16 5.959676e-09 1.077308e-16 2.733015e-

Residual Deviance: 1578.536
AIC: 1602.536
```

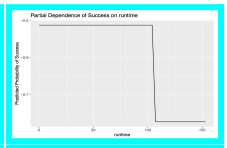
	Actual		
Predicted	High	Low	Medium
Low	0	199	40
Medium	0	26	15
High	6	23	52
Very High	17	20	41

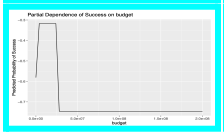
The accuracy of this model is about 50.11%. This model doesn't perform that great on new data. It is one of the worst performing models we have seen. This was meant to predict the budget based on revenue, release year, and popularity. Revenue had the highest positive influence while release year had a negative influence and popularity had a small influence.

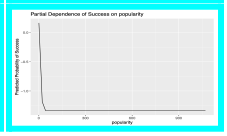
Classification with Emphasis on Interpretation:

Classifying Success with Decision Trees: The accuracy of this model is about 68.11%. This performs moderately well on new data. The decision tree structure shows how budget and popularity split the data to classify movies. The path from root to leaf highlights the decision rules. This easily identifies the most important features based on the splits. Based on the decision tree popularity is used to split at first, and then with release year splitting the data. Budget and runtime ends up splitting the data a little bit later in the decision tree process. While popularity and release year end up splitting

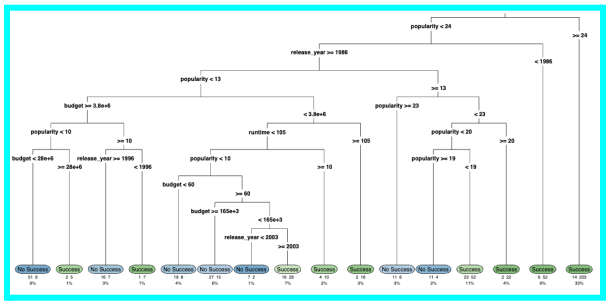






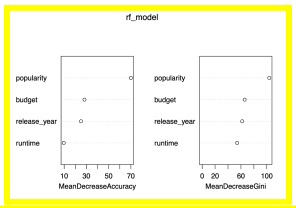


	Actual	
Predicted	No Success	Success
No Success	57	44
Success	96	242



the data farther down the tree. The PDPs show release year is the most important feature in the DTs.

Classifying Success with Random Forest (with Feature Importance): The accuracy is about 68.79%. This performs moderately well on new data. We can see popularity contributes the most to the model with budget next, then release year, and runtime being the least contributable predictor. The PDPs for this model shows that for each variable as the value gets higher the likelihood of success is higher. However, for popularity it tends to flatten out.



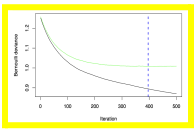
Actual		
Predicted	No Success	Success
No Success	57	41
Success	96	245

Actual		
Predicted	No Success	Success
No Success	32	25
Success	41	121

```
# weights:  31
initial  value 677.159206
iter  10 value 532.657456
iter  20 value 502.388653
iter  30 value 492.785404
iter  40 value 487.961420
iter  50 value 485.487982
iter  60 value 484.019104
iter  70 value 482.909123
iter  80 value 479.943907
iter  90 value 479.587481
iter 100 value 479.516698
iter 110 value 479.363997
iter 120 value 479.321355
final  value 479.319254
```

Classifying Success with Neural Networks: In the context of movie success classification, the neural network captures nonlinear relationships between features like budget and popularity interacting in unexpected ways. The accuracy of this model is about 69.86%. This model performs pretty well on new data and is one of the best performing models we have used so far. The model performed as expected and the loss function value went from 677.16 to 479.32. This means that the model has learned meaningful patterns in the data and stopped training at around a 479 loss value.

Classifying with Gradient Boosting: Gradient boosting combines weak learners (decision trees) into a strong predictive model. Using a Bernoulli distribution for binary classification, we determine the optimal number of trees through cross-validation. The model achieves 67.20% accuracy, performing moderately well on new data. It ranks popularity as the most influential variable, followed by release year, budget, and runtime—differing from other models, which rank budget second. The graph shows the Bernoulli deviance between predictions and actual values (green/black).

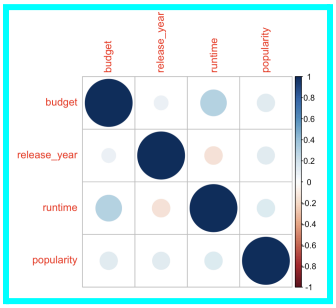


	var	rel.inf
popularity	popularity	49.77012
release_year	release_year	20.57941
budget	budget	19.09468
runtime	runtime	10.55579

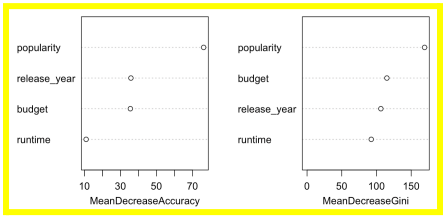
Actual		
Predicted	No Success	Success
No Success	54	45
Success	99	241

Feature Selection and Comparison of Predictor Sets: We will analyze the correlation between predictors and the target variable, success. We will identify multicollinearity among predictors to avoid redundancy. We will do so by looking at a correlation matrix for numeric predictors. We performed all below feature selection steps on all variables in the data set as well on the rmd file, and got similar results.

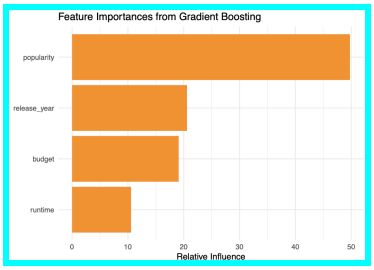
Correlation Analysis to check for Multicollinearity: Checking to make sure there is no multicollinearity present for the variables we chose to predict. We can see that there is no multicollinearity present as the correlations between the variables we chose are small.



Variable Importance from Random Forest: In the random forest model, popularity remains the most influential predictor of success. However, the second most influential variable changes depending on the metric used. MeanDecreaseAccuracy ranks release year next, followed by budget and runtime, while MeanDecreaseGini ranks budget next, followed by release year and runtime.



Variable Importance from Gradient Boosting: Shows popularity, then release year, then budget, then runtime is the order of relative influence of predictor variables to the overall success.



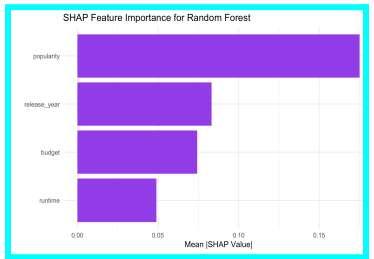
Stepwise Selection: We can use stepwise regression to identify the most relevant features for logistic regression. Release_year and popularity strongly influence movie success: success decreases as release_year increases, while higher popularity boosts success. Runtime has a weaker effect. Initially, the model included budget, release_year, runtime, and popularity. Stepwise selection removed budget, leaving release_year, runtime, and popularity. The AIC value improved slightly from 1313.79 to 1311.8, indicating a better fit without budget.

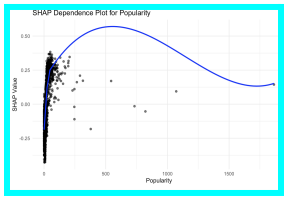
```
    Null deviance: 1396.4  on 1097  degrees of freedom
Residual deviance: 1303.8  on 1094  degrees of freedom
AIC: 1311.8

Number of Fisher Scoring iterations: 6
```

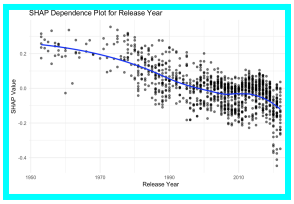
```
Start:  AIC=1313.79
success ~ budget + release_year + runtime + popularity

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

               Df Deviance    AIC
- budget        1   1303.8 1311.8
<none>              1303.8 1313.8
- runtime       1   1306.9 1314.9
- release_year  1   1338.3 1346.3
- popularity    1   1348.3 1356.3

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

Step:  AIC=1311.8
success ~ release_year + runtime + popularity

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

               Df Deviance    AIC
<none>              1303.8 1311.8
- runtime       1   1307.0 1313.0
+ budget        1   1303.8 1313.8
- release_year  1   1338.5 1344.5
- popularity    1   1350.1 1356.1
```

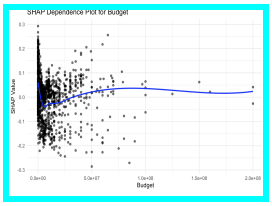
```
Call:
glm(formula = success ~ release_year + runtime + popularity,
    family = binomial, data = df_for_class)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  57.087853  10.267083   5.560 2.69e-08 ***
release_year -0.028542   0.005102  -5.595 2.21e-08 ***
runtime       0.004466   0.002492   1.792   0.0731 .
popularity    0.017716   0.003308   5.355 8.54e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

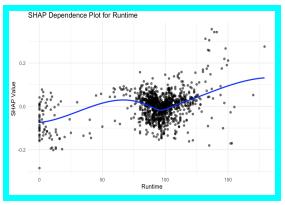
SHAP Values for Model Interpretability: Again, this predicts the same as the other models with order of importance being popularity, release_year, budget, runtime. Now we will plot a SHAP dependence plot for each variable to see the complete view of the model's decision-making.











Popularity: There is a non-linear relationship, where popularity initially positively contributes to predictions but becomes negative after a peak SHAP value of 0.5, indicating limited interaction effects with other variables. **Release Year**: Release year has a nearly linear negative relationship with predictions, with a maximum SHAP value of 0.2 and potential interactions suggested by the wider spread of SHAP values. **Budget**: Budget initially negatively impacts predictions but contributes positively when greater than 0, with non-linearity and a wide spread at lower values indicating interactions with other variables. **Runtime**: Runtime shows a near-linear positive relationship with predictions, with a maximum SHAP value of 0.12 and potential interactions indicated by the spread of SHAP values.

RFE for Feature Selection: This chooses popularity as the selected variable as popularity is the 4th variable inputted into the rfe function.

Variables <S3: AsIs>	Accur... <S3: AsIs>	Kappa <S3: AsIs>	Accuracy... <S3: AsIs>	Kappa... <S3: AsIs>	Selected <S3: AsIs>
1	1	0.6185	0.1732	0.03062	0.05753	
2	2	0.6776	0.2573	0.04788	0.10391	
3	3	0.7277	0.3740	0.02213	0.05877	
4	4	0.7331	0.3747	0.02120	0.06461	*

Multiple Regression with z-scores for Variable Importance: We can use Pr|z| to find statistically significant features. Since release year and popularity are the variables with p values less than 0.05, we can say these are the variables that are statistically significant at predicting success.

```
glm(formula = success ~ budget + release_year + runtime + popularity,
    family = "binomial", data = training_lg)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.35118    0.50679   2.666  0.00767 **
budget       -0.98439    1.00381  -0.981  0.32676
release_year -1.97016    0.47477  -4.150 3.33e-05 ***
runtime       0.05776    0.62504   0.092  0.92637
popularity   89.83537   13.20727   6.802 1.03e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 827.89  on 658  degrees of freedom
Residual deviance: 720.31  on 654  degrees of freedom
AIC: 730.31

Number of Fisher Scoring iterations: 6
```

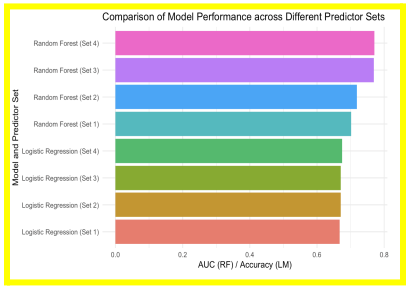
Comparing Models with Different Predictor Sets: We have one example of a reduced model above. Let's continue and keep comparing different predictor sets all including popularity since this was unanimously the most influential feature predicting success. We will train and evaluate models with different predictor sets for Random Forest and train and evaluate models with different predictor sets for Logistic Regression. We can see that using both models, set 4 (the one with all predictors) performs best, followed by set 3 (with only budget, release year, and popularity). This is in line with feature selection results, as it showed these are the 3 most influential predictors. Looking at the AIC and BIC results, set 2 is the best since it has the lowest results when taking a penalty for more predictors.

Adding Interaction Terms: We can test whether interaction terms improve model performance. We chose to interact budget and release year as the SHAP models suggested these were the two variables with potential interactions. Our interaction model include popularity + budget*release_year. With an accuracy of about 66.12%, this model does not perform better than the predictor sets above.

```
predictor_set_1 <- c("budget", "popularity")
predictor_set_2 <- c("release_year", "popularity")
predictor_set_3 <- c("budget", "release_year", "popularity")
predictor_set_4 <- c("budget", "release_year", "runtime", "popularity")
```

Predictor_Set <chr>	AIC <dbl>	BIC <dbl>
Set 1	1354.589	1369.593
Set 2	1313.046	1328.050
Set 3	1314.918	1334.923
Set 4	1313.794	1338.801

Model <chr>	AUC_or_Accuracy <dbl>
Random Forest (Set 1)	0.7020184
Random Forest (Set 2)	0.7188753
Random Forest (Set 3)	0.7690725
Random Forest (Set 4)	0.7708759
Logistic Regression (Set 1)	0.6675774
Logistic Regression (Set 2)	0.6703097
Logistic Regression (Set 3)	0.6703097
Logistic Regression (Set 4)	0.6748634



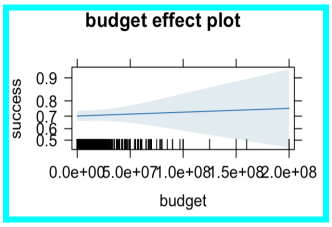

```
Interaction Model Accuracy:  0.6612022
```

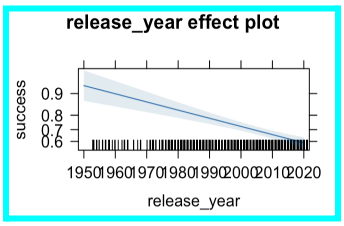
Cross-Validation to Compare Models: Use cross-validation to evaluate the generalizability of each predictor set. We chose to use the best performing sets from random forest and linear regression (set 3 and 4). Set 4 with all 4 variables is our base model, and set 3 is our reduced model. With a slightly higher kappa and accuracy, set 4 still performs best.

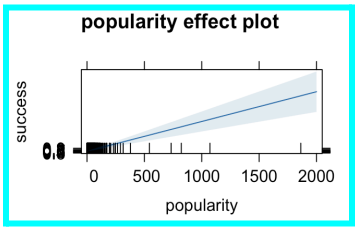
```
1098 samples
   4 predictor
   2 classes: 'No Success', 'Success'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 879, 878, 878, 879, 878
Resampling results:

  Accuracy   Kappa
  0.6748692  0.1067336
```

```
1098 samples
   3 predictor
   2 classes: 'No Success', 'Success'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 878, 879, 879, 878, 878
Resampling results:

  Accuracy   Kappa
  0.6711955  0.1034355
```

Effect Plots for Logistic Regression: Visualize the effect of individual predictors on the probability of success. All predictors show a somewhat linear relationship, indicating a constant rate of change in the log odds/probability of success. Budget is pretty constant, whereas the release year shows a negative relationship: as the year of release increases the probability of success decreases. Popularity has a positive relationship: as popularity increases the probability of success increases. Budget has the widest confidence interval (shaded region) meaning there is greater uncertainty in the estimated effect than popularity and release year.







Conclusion: The model that was most accurate at predicting success from features including budget, release year, popularity, and runtime was Neural Networks with an overall accuracy rate of 69.86%. The worst performing model at classifying a movie's success was LDA with an accuracy rate of 62.41%. Based on feature selection, the two most significant features were popularity and release year. When comparing predictor sets, the set with all 4 variables performed the best. However, this will likely be true most of the time because when you increase predictors, the model is more accurate. Since the AIC and BIC was lowest with model 2, and accuracy was only slightly worse, we conclude this is the best predictor model to use.

Citations:

1. Session 7 Lab: Airline Delays Case Study Notebook
2. Session 8 Lab: Credit Scoring Case Study Notebook
3. Session 9 Lab: Customer Churn Case Study Notebook
4. Mock, Thomas (2022). Horror Movies. Retrieved from https://github.com/rfordatascience/tidytuesday/tree/main/data/2022/2022-11-01

Disclaimer: We believe every model would have performed better if there was more data to represent. We chose to normalize instead of standardize due to the small dataset and because data was not normally distributed with skewed distributions for our variables of interest.