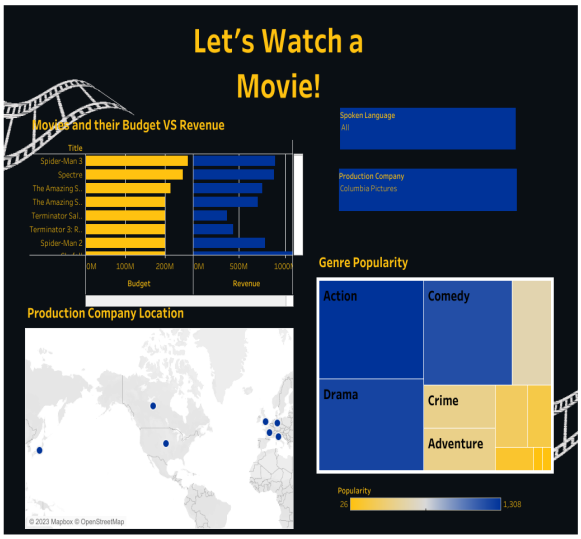
Executive Summary Group 1

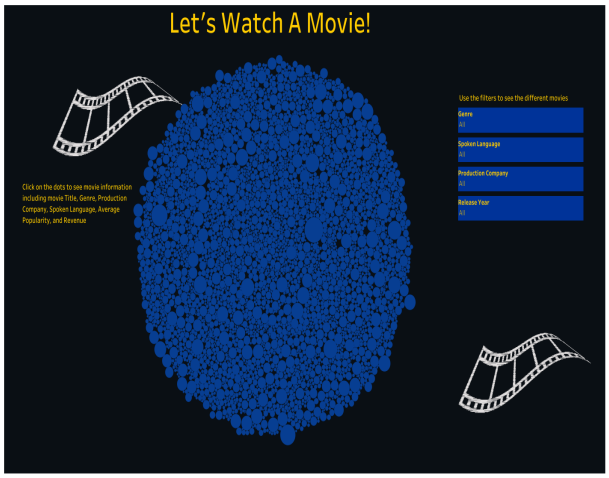
Alyssa DiFurio, Alexandra Flores, Bekhem Horne, Vanessa Martinez

In this project our goal is to create a machine learning search engine that will allow our users to create their own movie from a selection of menu items to see if the movie would be a success or not. We found our dataset from Kaggle, The Movies Dataset, which holds over 45,000 movies to build our machine with. Our inspiration came from the fact that as most people, we all collectively love to escape reality by going to see a movie. Our expectation of our search engine is to provide the user with a fun model to experience building their own movie, as well as see several visualizations of how the data works and compares.

Our original idea was to create a machine learning recommender that would allow the user to select actors, genres, release years etc. to find new movies that coincide with their selections. However, we found that the existing seven CSV files used in our Kaggle dataset need far more cleaning than time allowed. We are still using this original dataset, but we found that by using only the main CSV named “movies_metadata” we were able to clean the data according to our desired results easier and faster to still produce a great search engine webpage. The cleaning of this dataset consisted of parsing our data to be able to clean the columns that were lists of dictionaries, creating new column names to place the separated data into, correcting the types, dropping columns we won’t need, and lastly taking a sample and performing a .loc in the columns for genres, spoken languages, production company and production country. Once this data was cleaned we made a CSV from the Data Frame which was then used for our Tableau dashboards, as well as a sample of the data for our data source page to show the Data Frame table.

There are three Tableau dashboards shown on our webpage. The color choices are representative of the Blockbuster logo and a plain black background with white film strips, to not distract from the visualizations. Dashboard 1 shows all the movie titles comparing their budget versus their revenue, a map showing the production company locations, and a treemap depicting the popularity of each genre. This whole dashboard can also be filtered by production company and spoken language. Dashboard 2 is a movie title focused dashboard using a packed bubbles chart that can be filtered by genre, production company, language, and the year in which the movie was released. Dashboard 3 shows a bar chart to visualize the average amount of votes a movie title has had, as well as a line graph comparing the vote counts by the runtime of a film. This was meant to be used to see if runtime affects the amount of votes a movie has, as well as the year it was released. This dashboard can also be filtered by genre, production company, spoken language, the release year, and release month.





For our machine learning data, we used the cleaned CSV to begin. We selected our target for the model to be voter averages greater than a value of six before dropping remaining columns not needed for the machine learning and then performing get dummies on the remaining data. Next, we began testing models to see which would be the most accurate for our desired results. Viewing our machine learning notebook you will see the Logistic Regression model to have the worst accuracy, with an F1 score of 65%, however no signs of overfitting. We continued to try several other models, Decision Tree Classifier had severe overfitting, Random Forest Classifier also had overfitting, AdaBoost was the first to yield good accuracy at 71% with no signs of overfitting, XGB Classifier at 73% , no overfitting and a Brier score of 1414. Our last model however, LightGBM yielded the best results. An F1 score of 74% with no overfitting, AUC of .8177 and a Brier score of 1392. We decided between XGB Classifier and LighGBM that the LightGBM model was the clear winner for our model.

```
   METRICS FOR THE TESTING SET:
   ----------------------------
   [[2883 1011]
[1126 2985]]

                precision    recall  f1-score   support

       False       0.72      0.74      0.73      3894
        True       0.75      0.73      0.74      4111

    accuracy                           0.73      8005
   macro avg       0.73      0.73      0.73      8005
weighted avg       0.73      0.73      0.73      8005


AUC for the Model Test Set: 0.8177237726534982
```

With our model chosen we moved forward to look at the feature importance. This was somewhat surprising as our top importance was not what we expected to see. Voter count, release year and runtime to name a few were the top three. Based on these results we mapped out the selection features we wanted to provide the users with on our search engine to ensure the results from the users inputs could potentially yield good results. Finishing the machine

learning side of our project we created the form needed in our HTML code to build our search engine.

We would like our users to try to make as many movies as they want to see if they can make both bad and good movies and report back with the results. You will be surprised with what can make a movie good or bad. We look forward to seeing how well our page performs.