**INTRO**  
(Maggie - slides 1-5)

* (OPENING SLIDE)
* We believe everyone loves movies! Or at least everyone has an opinion about which movies are the best!
* (ADVANCE TO SLIDE 2 – popcorn icon)
* We picked movies as a topic since they’re our go-to escape from the ever-present pandemic.
* Streaming services use data modeling to suggest what to watch next and we wondered
* Who’s using data to determine what to make next?!
* (ADVANCE TO SLIDE 3 - white page on orange background)
* Introducing Movie MARK, a startup movie data company seeking funding.
* The Movie MARK team is comprised of Maggie, Andrew, Rose, and Kathy and we chose to answer the question-
* what makes a movie successful?
* (ADVANCE TO SLIDE 4 - black slide with white/orange text)
* But first we must define movie success
* (ADVANCE TO SLIDE 5 - Oscar award)
* For our purposes, we consider a successful movie as one that is
* Popular
* Profitable
* And perhaps won notoriety through industry awards.

**CHARACTERS**  
(Maggie - slides 6-9)

* (ADVANCE TO SLIDE 6 - Ellen Ripley)
* Before we dive into our data, let’s meet our stakeholders
* Meet Ellen.
* She’s is launching a streaming service and wants to make sure she’s delivering what her audiences want and maximizing her capital.
* (ADVANCE TO SLIDE 7 - Sam)
* Meet Sam.
* He got a few promotions and is now an up-and-coming movie exec ready to launch the next blockbuster hit.
* He thinks Movie MARK® can give his studio a competitive edge.
* (ADVANCE TO SLIDE 8 - Steel)
* Last but not least, meet Steel.
* His lovely new bundle of joy means his time is very limited.
* Being a big movie enthusiast, he wants maximize his movie watching time without wasting hours searching for what to watch next.
* (ADVANCE TO SLIDE 9 - spoiler alert!)
* Spoiler alert!
* Our proof-of-concept machine learning model was only able to evaluate our numerical variables around ratings prevented us from truly predicting a movie’s success.
* Let’s talk about why.

**DATA EXPLORATION**  
(Rose - slides 10-13)

* (ADVANCE TO SLIDE 10 – orange worldwide 1970+)
* To help our cast of 3 movie enthusiast stakeholders, we pulled movie data from two primary sources – ‘IMDb movies.csv’ and ‘movies\_metatdata.csv’ both from Kaggle.
* While we initially explored using a 3rd file - ‘film awards IMDb.csv’, which listed various global awards, their award’s categories and the associated winners, but it was identified in the initial exploration of the data in the files identified two common but difficult issue when working with big data; ‘variety’ and ‘veracity’
* These issues created such a large amount of scope creep, we had to make the difficult decision to eliminate the use of a third file, as cleaning our data in preparation for analysis already took us a solid 2 weeks even with the more robust IMDb and TMDB data-sets.
* (ADVANCE TO SLIDE 11 – list of data)
* Our files included:
* categorical variables like ‘Directors’, ‘Actors’, ‘Genres’, ‘Producers’, fields with plot keywords
* fields with numeric data like ‘budget’ and ‘revenue’,
* as well as popularity data, ranging from proprietary rating systems (metascore, for example) to user and critic – ‘ratings’ and ‘counts’
* and finally there was dates and times in fields such as ‘release dates’, ‘duration’ and ‘length of film’.
* (ADVANCE TO SLIDE 12 – unmanageable variety)
* We ended up eliminating some of the fields, such as actors, writers, directors and producers, because of the shear breadth of the data (the variety issue) whereas one field could be so vast in its data it was unmanageable.
* (ADVANCE TO SLIDE 13 – low veracity)
* Additionally, some of the fields were not well populated (veracity) such as (and surprisingly) budget and revenue
* (ADVANCE TO SLIDE 14 – missing budget data)
* which were missing 55-80% over-all proving to be an inadequate field for our modeling.
* (ADVANCE TO SLIDE 15 – orange funnel)
* In order to make the files readily available to the public, we stored our cleaned files ‘imdb\_ and ‘tmdb\_main.csv’ in a D3 bucket in Amazon Web Services.
* We then used Jupyter Notebook to access and retrieve the files on AWS and do some additional manipulation of the files before we write them to a relational database hosted on Amazon web services in postgres (using pgAdmin 4)
* (ADVANCE TO SLIDE 16 – first technology slide)
* we then merged the two databases using sqlalchemy in jupyter notebook outputting our final file which is also hosted by AWS in postgres.

**DATA EXPLORATION** continued  
(Andrew - slides 17-21 + dashboard)

* (ADVANCE TO SLIDE 17 – full technologies list)
* Real quick recap of other tech we used.
* Jupyter Notebooks, obviously significant.
* Call out: ProfileReport dependency.
* Data hosted in a PostgreSQL on AWS.
* Chose Tableau for versatility/plug ‘n play.
* (ADVANCE TO SLIDE 18 – release year heat map)
* We had priors about what could predict movie success. And a few of those features jumped out at us when we began poking around the raw and clean versions of the data.
* For instance, there seemed to be a fairly clear relationship between release year/engagement with a film, as depicted by the number of user reviews (internet, recency bias, expansion of IMDb/TMDB datasets? All plausible contributors)
* (ADVANCE TO SLIDE 19 – first scatter plot)
* Another one – average IMDb scores and TMDB scores of films. These are weighted user scores of each film.
* (ADVANCE TO SLIDE 20 – second scatter plot)
* That IMDb average also seemed to correspond with Metascore, a measure generated by critics’ input (similar lines as RT?)
* (ADVANCE TO SLIDE 21 – confusion matrix)
* Similar story told by correlation matrix.
* Not all of our variables were represented here, and you already know we didn’t end up keeping many of them.
* But perhaps just as interesting as positive correlations were weak/negative correlations.
* Release year – more engagement, more negativity with more new projects?
* Fewer high-quality projects?
* Nostalgia for old movies?
* (DASHBOARD)
* \*You can get a more granular feel for some of the data in our dashboard. 1st viz – average vote by duration/worldwide box office numbers.
* Spider-Man search: -- differing opinions, similar stratosphere at the box (dedicated fans of the franchise?)
* Fast & Furious (type: furious) – nice correspondence between well-liked/willingness to pay to see. Word-of-mouth factor?
* Both franchises: similar duration/runtime – with some exceptions, around the 120-130 sweet-spot
* Genres
* Biggest categories – drama and comedy. Interestingly, some of our modeling work suggested drama and animation were the best predictors for success. Drama/biggest piece of the pie. Animation/one of the smallest (not nearly the overlap in animation)
* Average Vote by Release year:
* Just quickly circle back to that potential, negative correlation between release year and scores could be due to drop-off in number of recent projects – data is not updated/pandemic releases held back, etc.)
* Director Success
  + We explored the director category as a variable, creating features based on the number of film credits and number that qualified as a ‘success’ based on our definition of the term. Ultimately, as Rose mentioned, it was too difficult to liberate and use as a feature in this model.
* TAKEAWAY:
* There were plenty of interesting pieces to the data. But just because something was interesting didn’t necessarily mean it was viable for our prediction task.

**DATA MODEL**  
DATA EXPLORATION continued  
(Kathy - slides 22-32)

* (ADVANCE TO SLIDE 22 – data issues)
* So, as I think we've made clear, we ran into the usual issues with web-scraped data.
* We ended up with three variables: genres, duration of the movie, and release year for about 70,000 movies.
* (ADVANCE TO SLIDE 23 – machine learning)
* Similar story on the success metrics IMDb score, was the only viable option.
* We used IMDb score >= 7 as our definition of success.
* So, 17% of our sample was labeled success
* (ADVANCE TO SLIDE 24 – model comparison)
* Since imdb score is a value between 1 and 10, one approach might be a linear regression model. But we wanted to discriminate between successes and non-successes, so a classification approach provided more options.
* (ADVANCE TO SLIDE 25 – deep learning call out)
* We started with a deep learning model because it would be tolerant of skewed distributions on the input variables, as well as some of the outliers in our data.
* The results were under-whelming, so we looked at other models.
* (ADVANCE TO SLIDE 26 – call out of models)
* We looked at random forest, logistic regression and support vector machine,
* (ADVANCE TO SLIDE 27 – accuracy)
* All showed accuracy around 80%. But that's not the best measure when you are more interested in the less plentiful successes. Let's look specifically at successes
* (ADVANCE TO SLIDE 28 – precision)

As you will recall, Precision tells movie producer Sam, if he backed a movie  
based on the model's prediction, how likely is it he backed a winner. These

* probabilities are pretty low.
* (ADVANCE TO SLIDE 29 – recall)
* Recall tells Sam if he has a winner, how likely is it that the model will recognize it. These low percentages mean an investor would miss a lot of great opportunities.
* (ADVANCE TO SLIDE 30 – F1)
* And low f1 means precision and recall are not in balance.
* (ADVANCE TO SLIDE 31 – EPOCH Graphs)
* And we looked at different iterations of the deep learning models and learned something interesting.
* We used a validation dataset, which was 30% of the data stripped from the training dataset.
* This allows us to over the course of the learning process, how well the model was learning on the test datasets, but also whether it was working on the validation dataset.
* This is useful in assessing model fit.
* And the surprising thing is this:
* These are results for two deep learning models identical except for the random seed used in selecting the train and test datasets.
* The green line shows the validation data. The blue line is the train/test data
* Notice that the loss curves on the right show the bottom model fitting much better than the other.
* The accuracy curves on the left show the validation data is consistently less accurate.
* So the training data is not representative of the validation data
* It turns out that the sci-kit learn algorithm selects the validation data by taking the last 30% of the training data by index.
* In our case the index is imdb\_id.
* So our validation dataset is more recent releases than the training dataset and as we know from our modeling efforts, release year carries a lot of weight in the models

**SUMMARY**  
(Maggie slides 32 - 38)

* (ADVANCE TO SLIDE 32 – the plot thickens)
* The plot thickens- having a ton of data to start doesn’t mean it’s useful.
* (ADVANCE TO SLIDE 33 – black “where to go from here”)
* So plot twist- Despite our data challenges, we learned a lot.
* While there’s a lot of possibilities on where to go next, we have a few listed here.
* We’d deep dive into the categorical variables to extract meaning in a systematic numeric way.
* Also, we thought natural language processing of reviews could identify new features to consider.
* Ultimately, our goal would be to build a model for predicting future success.
* The current approach classifies winning movies but it’s not designed to predict future success.
* That said, there’s not much we’d do differently.
* We worked great together, had a good division of labor and did the best we could with the data we had.
* Only thing we might have done differently is cloned this clever crew to double our capacity and maybe try a different topic.
* (ADVANCE TO SLIDE 34 – our characters, ellen is sad)
* So for our 3 characters, our proof of concept does not yet help Ellen launch her streaming service with a differentiated pricing model for successful movies.
* (ADVANCE TO SLIDE 35 – Sam is sad)
* Also, movie Exec Sam will have to wait to use machine learning to guarantee  blockbuster after blockbuster.
* (ADVANCE TO SLIDE 36 – but Steel is thrilled)
* However, we can proudly let Steel know to target dramas and animated movies rated 7-8 on IMDb if he wants to maximize his time watching movies, rather than searching for them.
* (ADVANCE TO SLIDE 37 – fin)
* Thank you, everyone