Stats 418 Final Project Report

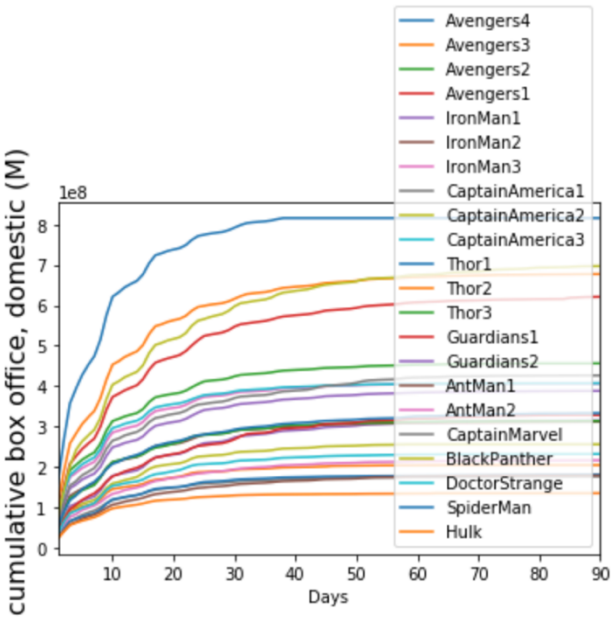
Lilian Gao (205061964)

Introduction

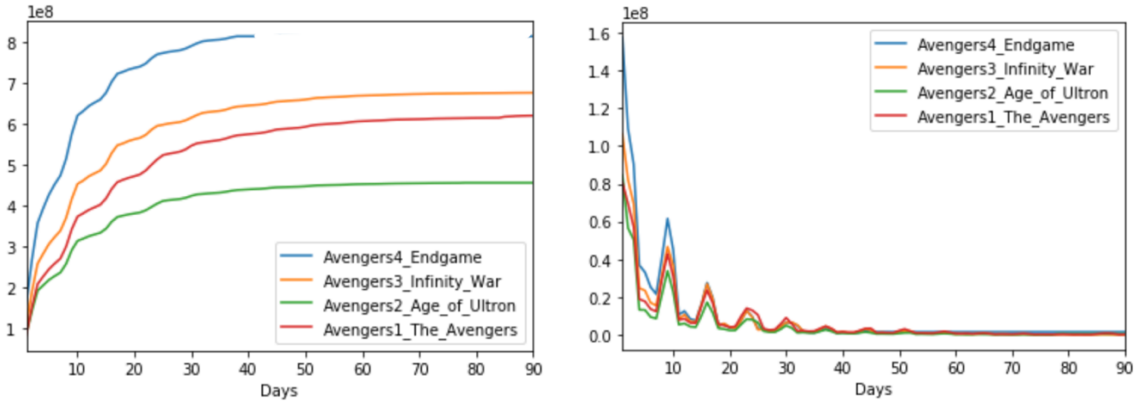
Avengers: Endgame is one of the biggest blockbusters in 2019. It was originated from the Marvel Cinematic Universe which has been well-loved over the decade. In this project, we will collect massive data from *the-numbers.com* which contains domestic box-office records of all this franchise movies so far. After aggregation and analysis, we would look more into the similarities among all those movies and develop a prediction model that can be applied and reproduced to forecast for the latest Avengers movie or even MCU movies and it should be easily accessible and reproduced.

Data Collection & Exploratory

Raw data were web-scrapped directly from each movie page under the tab "Box Office" on *the-numbers.com*. Python Requests and BeautifulSoup are two major packages used during this process. Since box offices mostly stop increasing after 3 months, I chose the first 90 days box-office from all except from the two new ones.



From the overall cumulative plot on the left, we can easily see that four out of five top box-office MCU movies are Avengers series. I found it interesting to see statistically how "Wakanda Forever" took over domestic market last year. Black Panther is the only top-selling non-Avengers movie and it shows rare and strong increasing trend until 70 days after its opening.

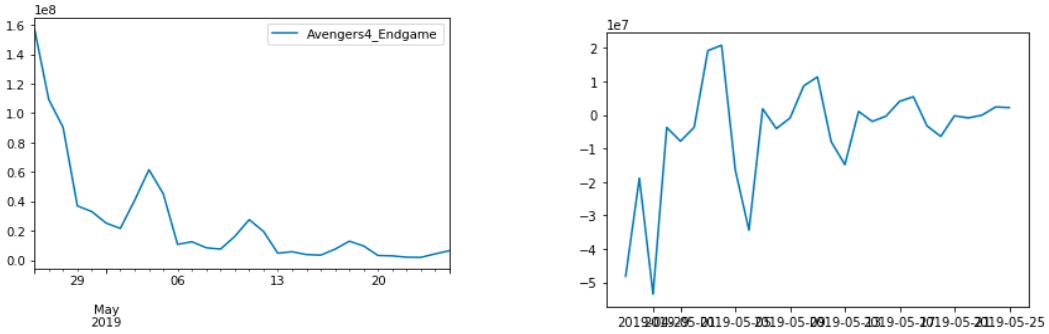


(Plots: Total Gross VS Daily Gross)

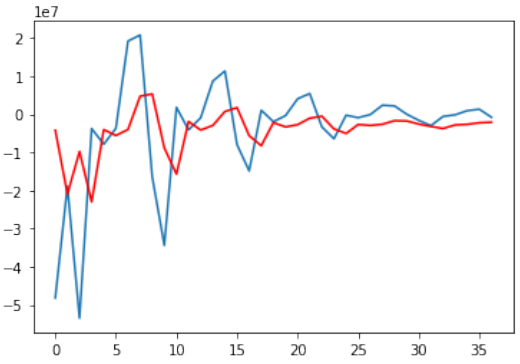
Therefore, since I'm more curious about how Endgame would perform, I zoomed in on the four Avengers movies only because they seem to have very similar trend and behavioral patterns. They all boosted rapidly during the first month and then slowly smoothed out. These may indicate that there may be some correlations among these 4 movies and I planned to randomly choose one as one of test samples for future analysis.

Data Analysis & Model Fitting

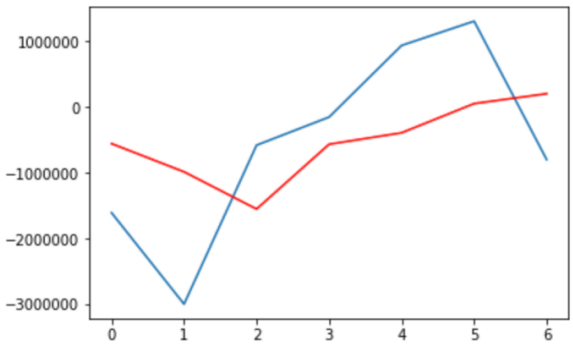
Since box-office are highly time-related, using Time Series becomes my major tool to develop a prediction model for Endgame. I took the first thirty days since its debut as a training set and the rest as a test set. After seeing the decomposition plot of the data with a clear trend of decreasing, I took differences on Endgame data to detrend and let it be much more stationary.



The function "auto_arima" was imported from pyramid.arima and was used to compare how different ARIMA model fits. Apparently, ARIMA(1,0,0) fits the best with smallest AIC and BIC. The following two plots shows that while the model ARIMA(1,0,0) fits the training set very well, it's not performing as accurately on the test data. This is something that can be improved with higher-level models in the future studies.



(av4 train: blue – actual, red - fitted)



(av4 test: blue – actual, red - fitted)

Reproduction

Prediction models are built and hosted on Amazon EC2 to make it accessible and applicable.

To reproduce this API, first pull and download the files in this repository.

Open Terminal, navigate to the docker directory, **then run:** `docker-compose up --build`

Open a new terminal under the same directory and run the curl command: `curl http://localhost:5000/` You should get a response: `server is up - nice job!`

To make a prediction of `box office` using `days` , run the following command as an example: `curl -H "Content-Type: application/json" -X POST -d '{"days":"1"}' "http://ec2-18-191-222-50.us-east-2.compute.amazonaws.com:5000/predict_endgame`

The return should be a data frame as
`dgross tgross`

`0 3868665.86566557 795931445.86566556`

`1 3249656.1819887 799181102.0476543`

The predictor values can be changed and will result in different prediction returns.

To stop your server running the API, type `ctrl-C` . Check to see if you have any docker containers running using `docker container ls` , and stop them through `docker container kill <container-name>` .

Conclusion & Future Ideas

The time series model ARIMA(1,0,0) seems to fit the data very well if applied within a month since the movie came out, but the accuracy decreased afterwards. It's reasonable because from the daily-gross plot we can see a dramatic change of pattern after 30 days and it may be related to a decrease of showtime in theaters as well as the influence and reviews from the critics and online scores. It would be interesting to see how these factors can be combined into time series models.