MILESTONE I REPORT

Team ID: CS_11

Youssef Khaled Abdallah

20191700767

Youssef Ashraf Eissa

20191700761

Youssef Khaled Farouk

20191700768

Omar Hany Abdelfattah

20191700420

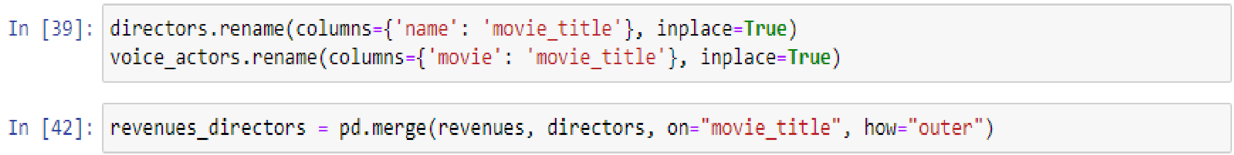
Amr Mahmoud Zakaria

20191700429

Milestone I Report

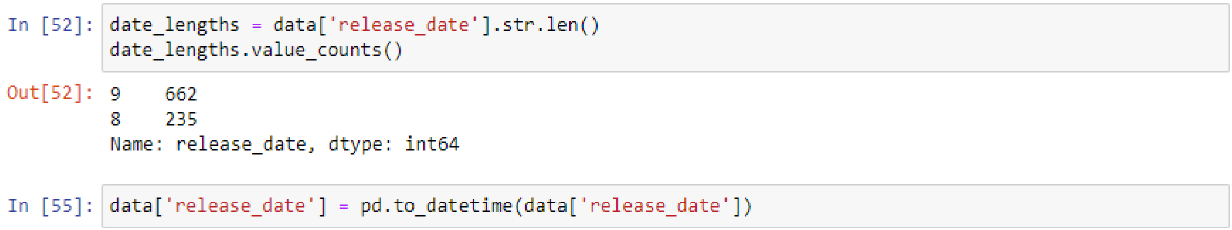
- **Preprocessing Techniques:**

  1. **Joining Data:** Using pandas outer merge method we joined the three separate data frames into a single data frame to perform preprocessing.



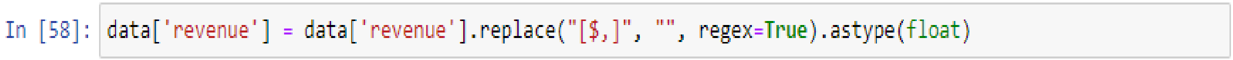
```
In [39]: directors.rename(columns={'name': 'movie_title'}, inplace=True)
         voice_actors.rename(columns={'movie': 'movie_title'}, inplace=True)

In [42]: revenues_directors = pd.merge(revenues, directors, on="movie_title", how="outer")
```

  2. **Parsing Dates:** Changed the data type of the “release date” column from Object to Date Time.



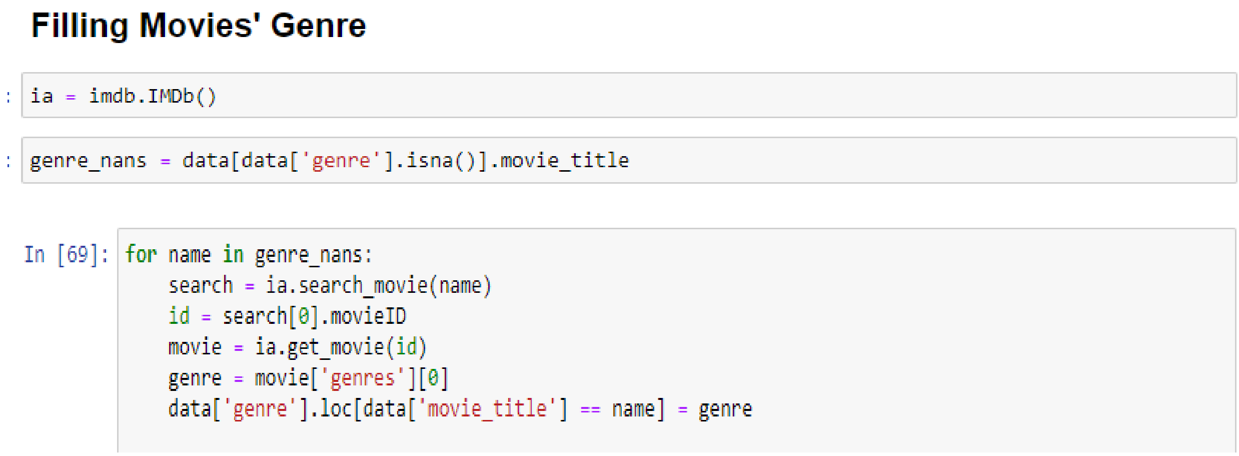
```
In [52]: date_lengths = data['release_date'].str.len()
         date_lengths.value_counts()

Out[52]: 9    662
         8    235
         Name: release_date, dtype: int64

In [55]: data['release_date'] = pd.to_datetime(data['release_date'])
```

  3. **Parsing Revenue:** Changed the data type of the “revenue” column from Object to float and removed the dollar sign ($) and commas using regular expressions.



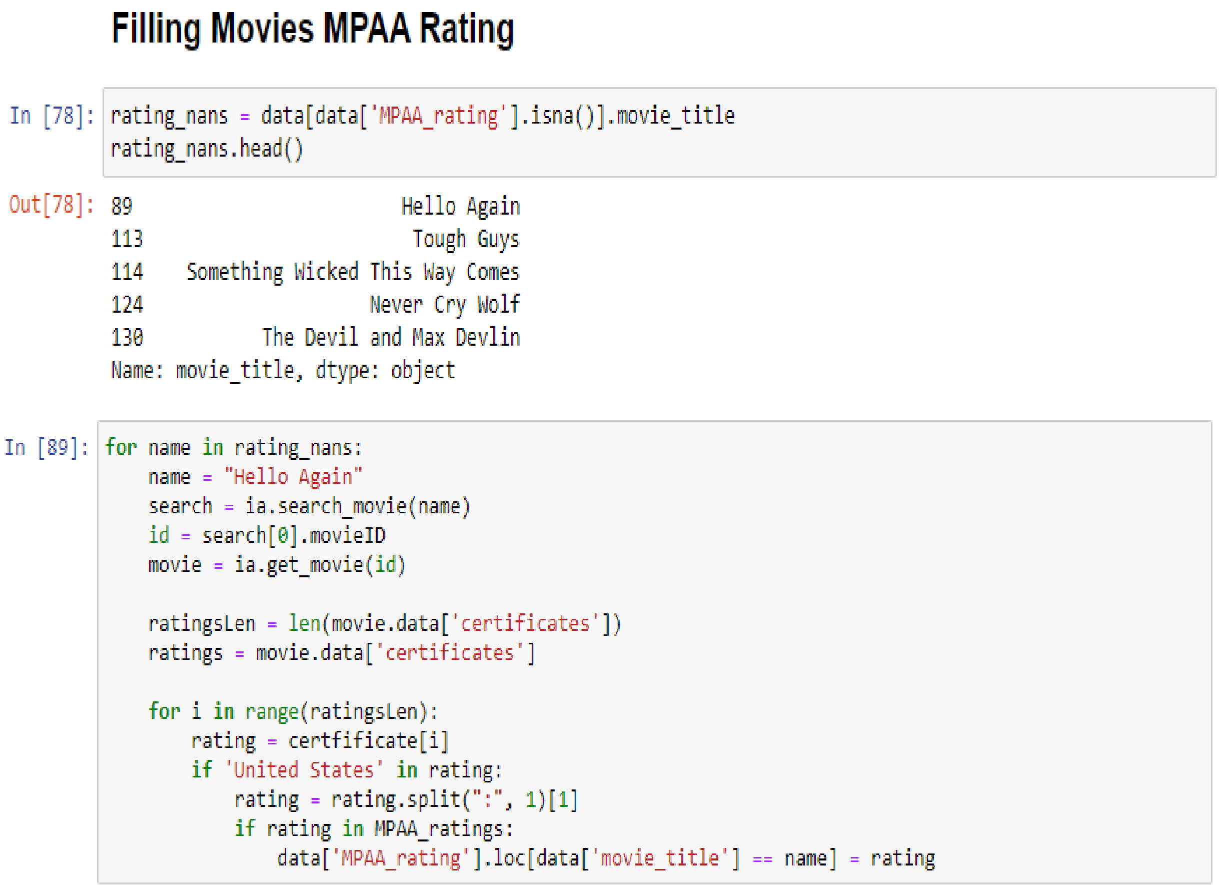
```
In [58]: data['revenue'] = data['revenue'].replace("[$,]", "", regex=True).astype(float)
```

  4. **Filling Nulls:** Filled nulls in “Genre” and “MPAA Rating” columns using ImbdPy Library.



Filling Movies' Genre

```
: ia = imdb.IMDb()

: genre_nans = data[data['genre'].isna()].movie_title

In [69]: for name in genre_nans:
             search = ia.search_movie(name)
             id = search[0].movieID
             movie = ia.get_movie(id)
             genre = movie['genres'][0]
             data['genre'].loc[data['movie_title'] == name] = genre
```

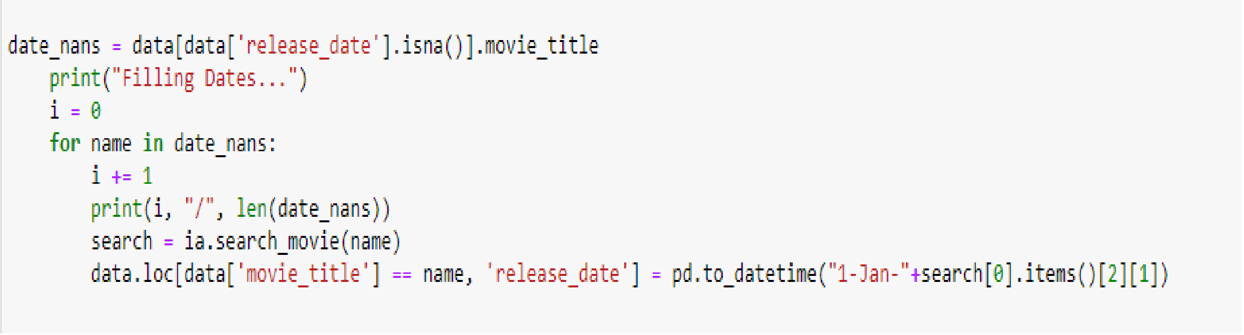


Filling Movies MPAA Rating

```
In [78]: rating_nans = data[data['MPAA_rating'].isna()].movie_title
         rating_nans.head()
```

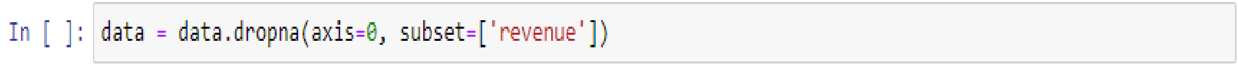
```
Out[78]: 89                          Hello Again
         113                          Tough Guys
         114     Something Wicked This Way Comes
         124                      Never Cry Wolf
         130               The Devil and Max Devlin
         Name: movie_title, dtype: object
```

```
In [89]: for name in rating_nans:
             name = "Hello Again"
             search = ia.search_movie(name)
             id = search[0].movieID
             movie = ia.get_movie(id)

             ratingsLen = len(movie.data['certificates'])
             ratings = movie.data['certificates']

             for i in range(ratingsLen):
                 rating = certficate[i]
                 if 'United States' in rating:
                     rating = rating.split(":", 1)[1]
                     if rating in MPAA_ratings:
                         data['MPAA_rating'].loc[data['movie_title'] == name] = rating
```



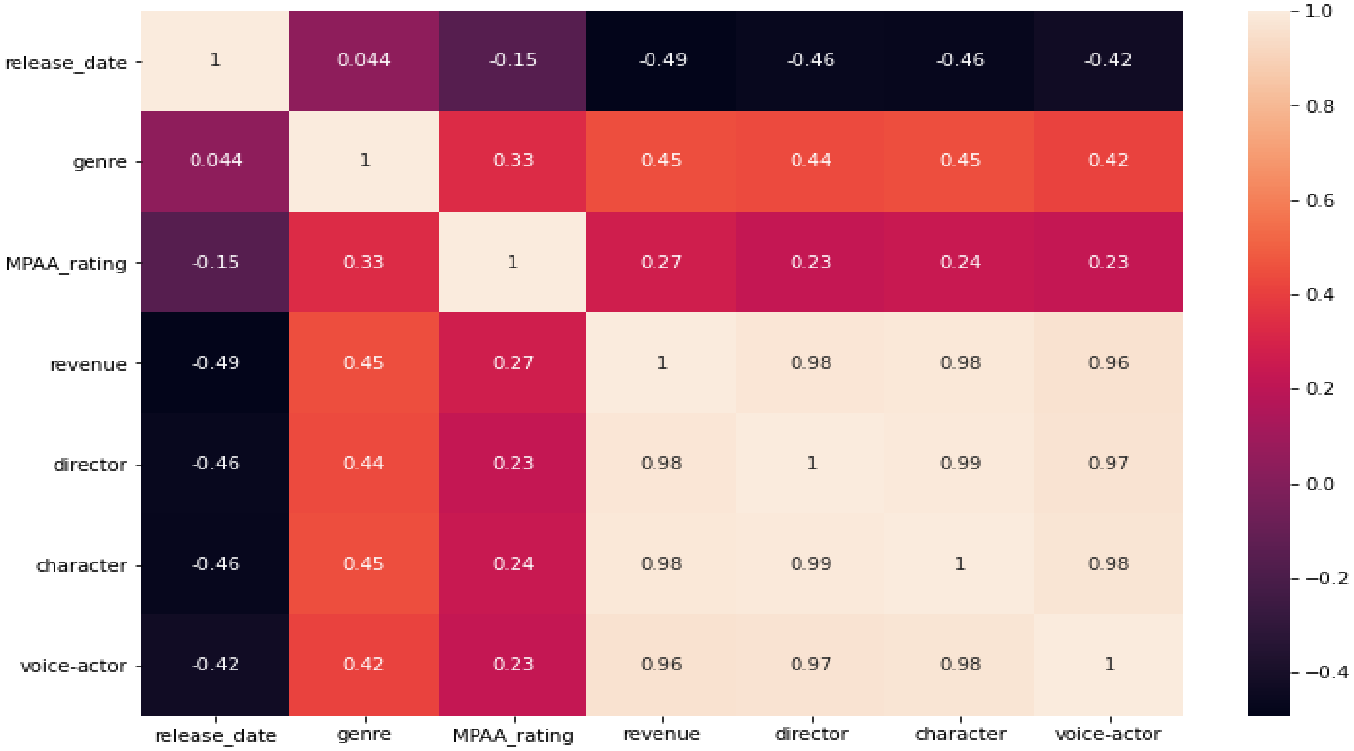
```
date_nans = data[data['release_date'].isna()].movie_title
    print("Filling Dates...")
    i = 0
    for name in date_nans:
        i += 1
        print(i, "/", len(date_nans))
        search = ia.search_movie(name)
        data.loc[data['movie_title'] == name, 'release_date'] = pd.to_datetime("1-Jan-"+search[0].items()[2][1])
```

5. Dropping Nulls: Dropped nulls in the "Revenue" column.

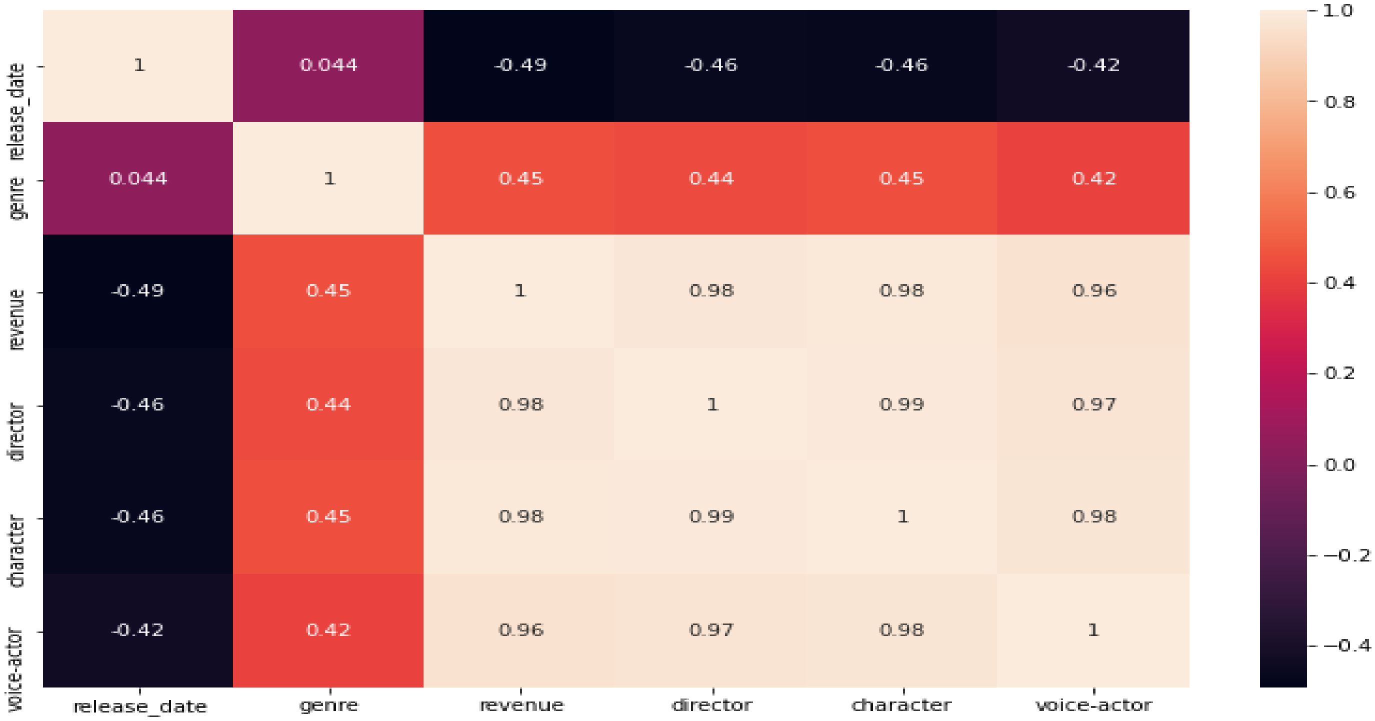


```
In [ ]: data = data.dropna(axis=0, subset=['revenue'])
```

- **Feature Analysis:**
  - Correlations between all the features in the dataset:



  - Correlations between all the features in the dataset after feature selection:



- **Regression Techniques:**
  1. Polynomial Regression.
  2. Multivariate Regression.

- **Differences between Models:**
  1. **MEstimate Encoder MODEL.py:**
     - Best Model Achieved.
     - Uses Polynomial Regression with degree = 3.
     - Uses MEstimate encoder technique.
     - Model Results:

```
Train Time: approximately 20ms.
Train MSE = 9240015442958470.0
Train Accuracy = 97.6837 %

Test MSE = 3808094164911872.5
Accuracy = 99.2290 %
```

  2. **TargetEncoder MODEL.py:**
     - Uses Polynomial Regression with degree = 2.
     - Uses Target encoder technique.
     - Model Results:

```
Train Time: approximately 20ms.
Train MSE = 1.6439359058938928e+16
Accuracy = 95.8789 %

Test MSE = 9906396911845948.0
Accuracy = 97.9944 %
```

3. JamesStein Encoder MODEL.py:

- Uses JamesStein encoder technique.
- Uses Multivariate Regression.
- Model Results:

```
Train Time: approximately 20ms.
Train MSE = 1.1130964506568598e+16
Accuracy = 97.1679 %

Test MSE = 1.6639074490577968e+16
Accuracy = 96.7837 %
```

- **Feature Selection:**

Used	release_date	genre	director	character	voice_actor
Dropped	movie_title	MPAA_rating			

- Features with correlation less than 30% are dropped.

- **Train Test Split:**
  - **Train Size:** 80%.
  - **Test Size:** 20%.

- **Improvements Techniques:**
  - Using ImdbPY Library we filled the nulls in the data set columns (eg. Genre, MPAA_rating, release_date).

- **Conclusion:**
  - This phase is about Regression techniques and how we use features to predict targets based on the correlations between features with each other and the target variable to get the best model to fit the data with least error and highest accuracy avoiding overfitting and underfitting.