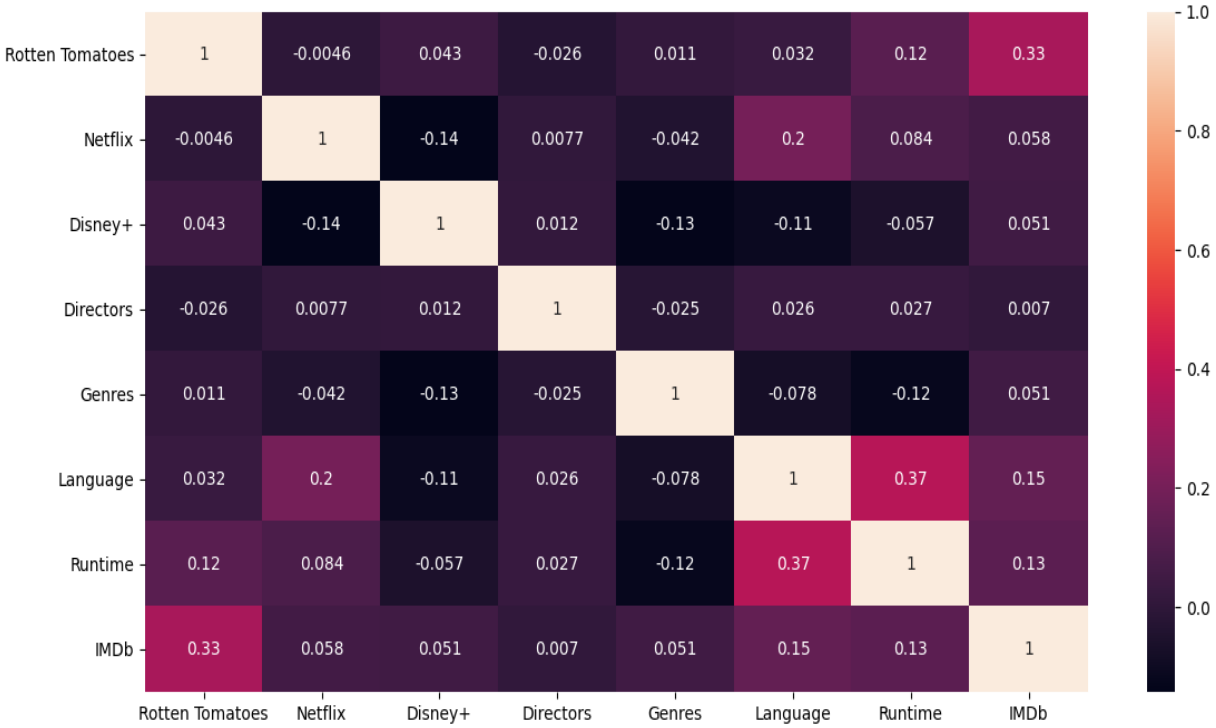
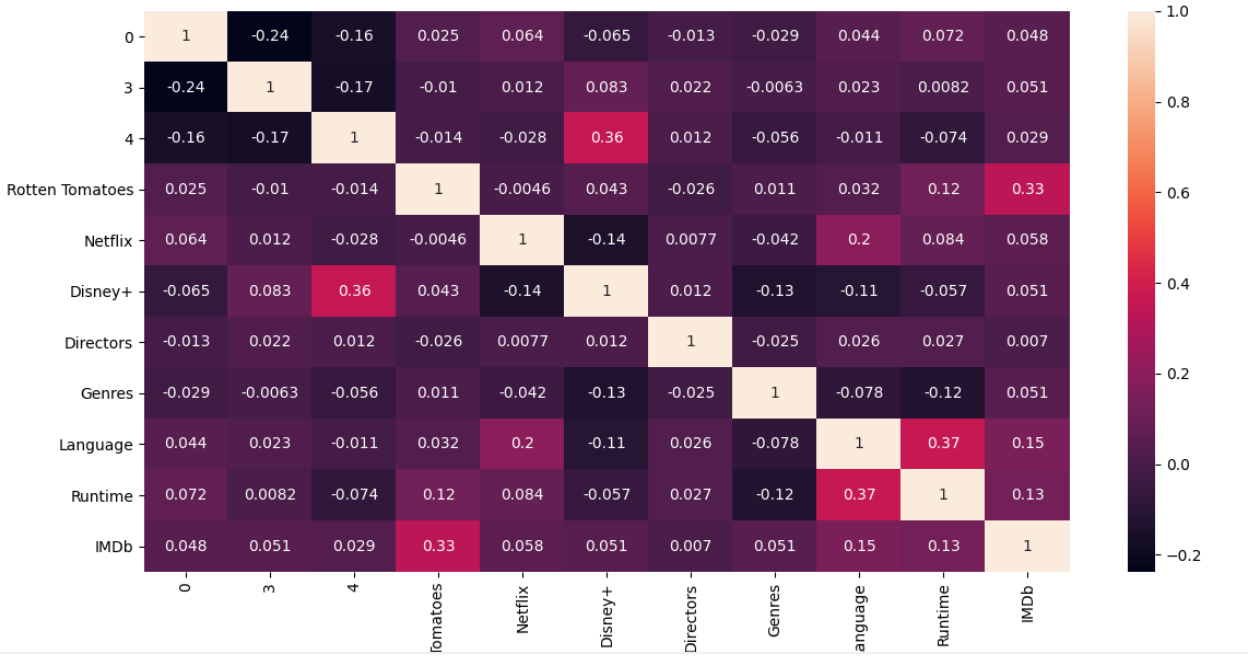
Report Predict Movie Success

1. preprocessing techniques
    - solve missing data by:
        - ➢ interpolate for (Rotten Tomatoes, age).
    - encoding for categorical data (6 columns):
        ['Directors', 'Genres', 'Country', 'Rotten Tomatoes', 'Language',’age’]
        - OneHotEncoder for age →the least one of different values.
        - OrdinalEncoder for the rest .
    - Normalization.
    - Drop columns that does not affected in result
    - Split data for train and test.
2. Correlation:





3. Regression techniques:

- **Multi Linear Regression:** It is the basic and commonly used type for predictive analysis. It is a statistical approach to modelling the relationship between a dependent variable and a given set of independent variables. Multiple Linear Regression attempts to model the relationship between two or more features and a response by fitting a linear equation to observed data.
  Y = b0 + b1 * x1 + b2 * x2 + b3 * x3 + …… bn * xn
  Y = Dependent variable and x1, x2, x3, …… xn = multiple independent variables
  Step #1: Data Pre-Processing
  Step #2: Fitting Multiple Linear Regression to the Training set
  Step #3: Predicting the Test set results.
- **Polynomial Regression:** (second degree) Polynomial Regression is a powerful technique to encounter the situations where a quadratic, cubic or a higher degree nonlinear relationship exists.
  $Y = b0 + b1 * x + b2 * x^2 + b3 * x^3 + \ldots\ldots bn * x^n$

4. **differences between each model:**
   - Mean Square Error:
     Multi Linear Regression: 0.019696934664050706
     Polynomial: 0.017397115586387072
   - Time for training:
     Multi Linear Regression: 10 sec.
     Polynomial: 7 sec.
5. **Features in models:**
   Title has been dropped because it has the least correlation with IMDb.
   Used:
   ['year','age', 'Rotten Tomatoes', 'Netflix', 'Hulu', 'Prime Video', 'Disney+', 'Type', 'Directors', 'Genres','Country', 'Language', 'Runtime'].

6. Size of Data:
   - train 7419
   - test 3180
7. further techniques:
   - OneHotEncoder.
   - OrdinalEncoder.
   - Remove missing values.
8. Screen shoot:

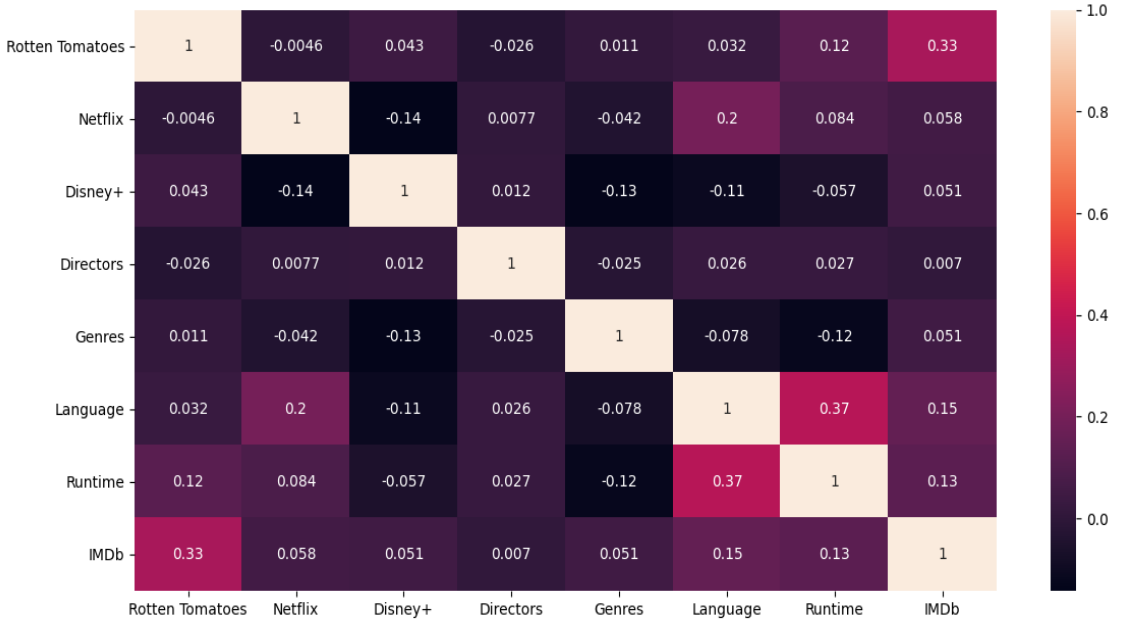
```
Mean Square Error Multi Linear  0.019696934664050706
Mean Square Error Polynomial 0.017397115586387072
```

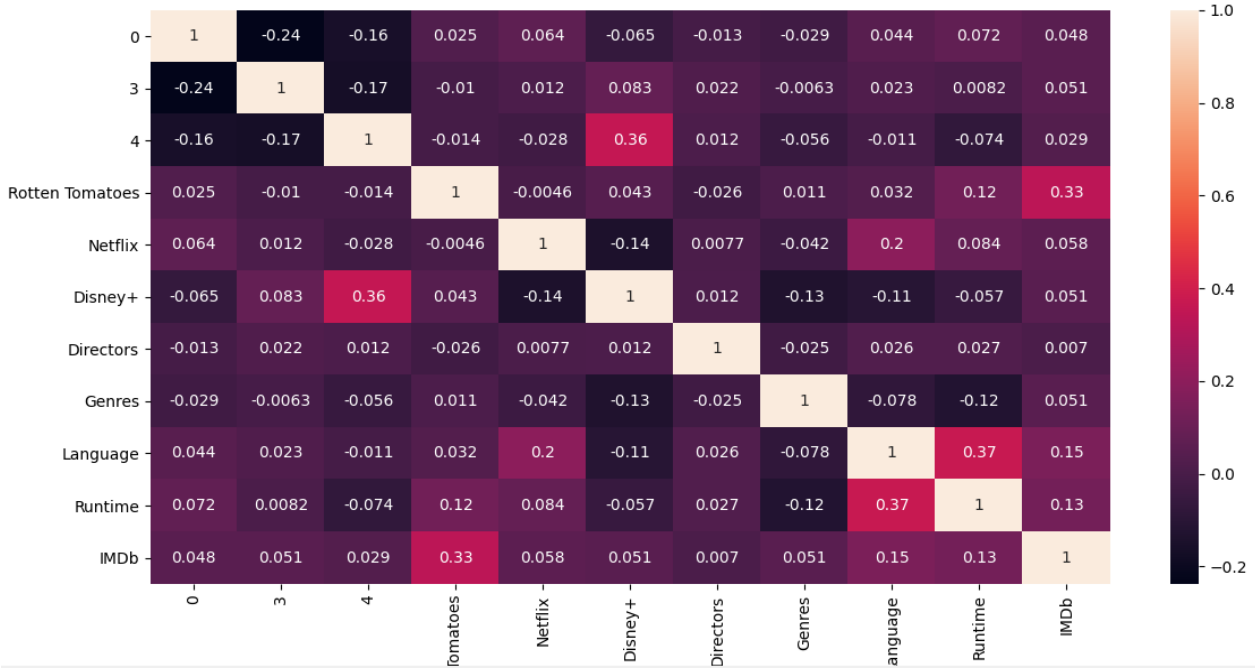
9. Conclusion:
   When use onehotencoding(‘age’)
   Polynomial at 2 degree is better than multi linear in error,
   Multi Linear Regression: 0.019696934664050706
   Polynomial: 0.017397115586387072
   When use onehotencoding('Rotten Tomatoes')
   Multi Linear Regression: 0.016
   Polynomial: size of columns is too large and not calculated.

For more Data (flow of code & flow chart**).**

Flow of code:

1. Drop 1 columns that have the most missing values and the title (doesn't affect the result).
    - According correlation





- Drop (title).

```
data = data.drop(['Title'], axis=1)
```

- Drop the rows that contain missing values (value = NULL).

```
data.dropna(how='any', inplace=True,
            subset=['Runtime', 'Language', 'Country', 'Genres', 'Directors', 'IMDb'])
```

2. Map string values for numbers.
   - preprocessing OrdinalEncoder:

```
ord_enc = preprocessing.OrdinalEncoder()
for i in ['Directors', 'Genres', 'Country', 'Rotten Tomatoes', 'Language']:
    data[i] = ord_enc.fit_transform(data[[i]])
```

   - Preprocessing OneHotEncoder

```
label_encoder = label.LabelEncoder()
data['Age'] = label_encoder.fit_transform(data['Age'])
onehot_encoder = preprocessing.OneHotEncoder()
sq = onehot_encoder.fit_transform(data[['Age']]).toarray()
```

3. Normalize data (fit data to be in the same scale).

```
data = pd.DataFrame(preprocessing.MinMaxScaler().fit_transform(data))
```

4. Divide data into input and output.

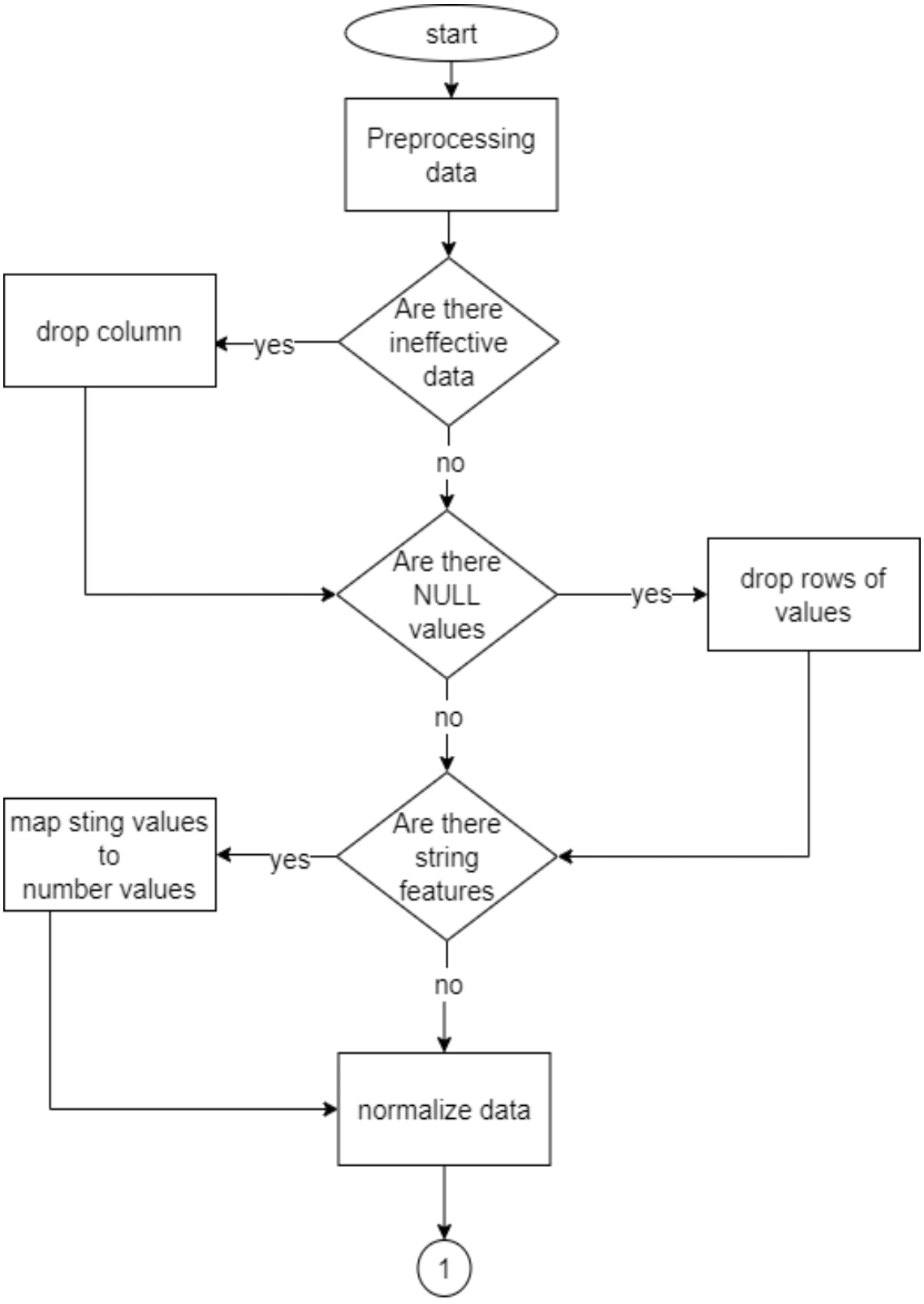
```
X = data.iloc[:, :-1]
Y = data.iloc[:, -1]
```

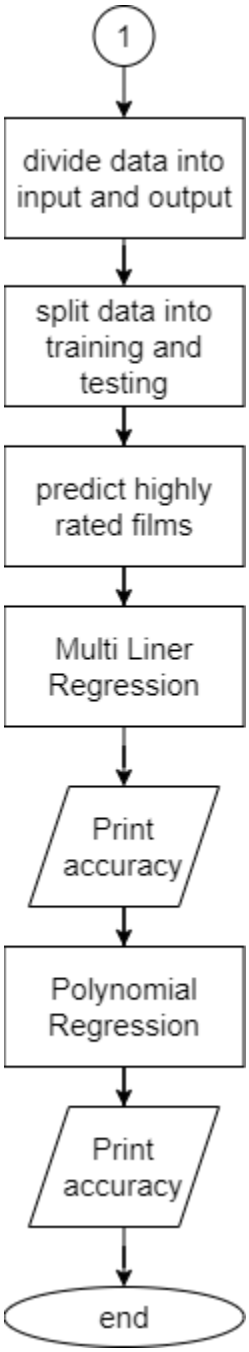
5. Split data set into training and testing.

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.30)
```

6. Regression:
   1. Multiple Linear Regression
      - Mean Square error = 0.019696934664050706
   2. Polynomial Regression
      - Mean Square error = 0.017397115586387072

Flowchart:


start
Preprocessing data
Are there ineffective data
yes
drop column
no
Are there NULL values
yes
drop rows of values
no
Are there string features
yes
map sting values to number values
no
normalize data
1


1
divide data into input and output
split data into training and testing
predict highly rated films
Multi Liner Regression
Print accuracy
Polynomial Regression
Print accuracy
end