What movie do you want to watch next?

predicting movie success using Movie MARK®

Everyone loves movies.

Streaming has reshaped cinema and the COVID-19 pandemic has left many of us wondering “what should we watch next?”

Using data-wrangling, programming, and machine learning skills, we plan to answer:

What makes movies successful?

Movie MARK®

predicting movie success

Our MARK® Team

of budding Data Scientists collaborated virtually across Zoom and Slack (as "the_clever_crew") to bring you this fine work.

- **Maggie Allen**
  Presentation + GitHub + Dashboard
- **Andrew Malony**
  GitHub + Graphs + Dashboard
- **Rose Baumann**
  Database + Data Clean-up
- **Kathy Morrissey**
  Data ETL + Machine Learning Model

But first we must ask
How do we define Movie Success?

suc·cess

/səkˈses/

noun

1. **Popularity** (Proprietary Scores, User Ratings, Critic Ratings)
2. **Estimated Profitability** (Revenue-Budget)
3. **Awards**

Meet Ellen.

She is the owner of a new start up streaming service, Serenity Streaming. She's looking to use AI and Machine Learning to help connect users with their favorite movie they have never even heard of.

Right now she's still working out of her home office and realizes despite a ton of data, she needs a proof of concept machine learning model to get investment interest.

Meet Sam.

Sam is a newly promoted executive at ABC Movie Productions and is interested in determining the right mix of movie genre, Director's talents, and A-list actors are going to be the recipe for the next blockbuster.

Before him, the boomers were sitting in rooms making all the calls but he thinks data science can flip the script.

Meet Steel.

He recently had a baby so he has no time to watch a bunch of bad movies. When he finally has a free evening, he wants the first movie he streams to be one he's happy to talk about with his new baby boy.

Let's see what he should look for in his next popcorn night's entertainment...

Spoiler Alert:

Our proof of concept was limited to analyzing ratings numeric variables.

Our Data Exploration

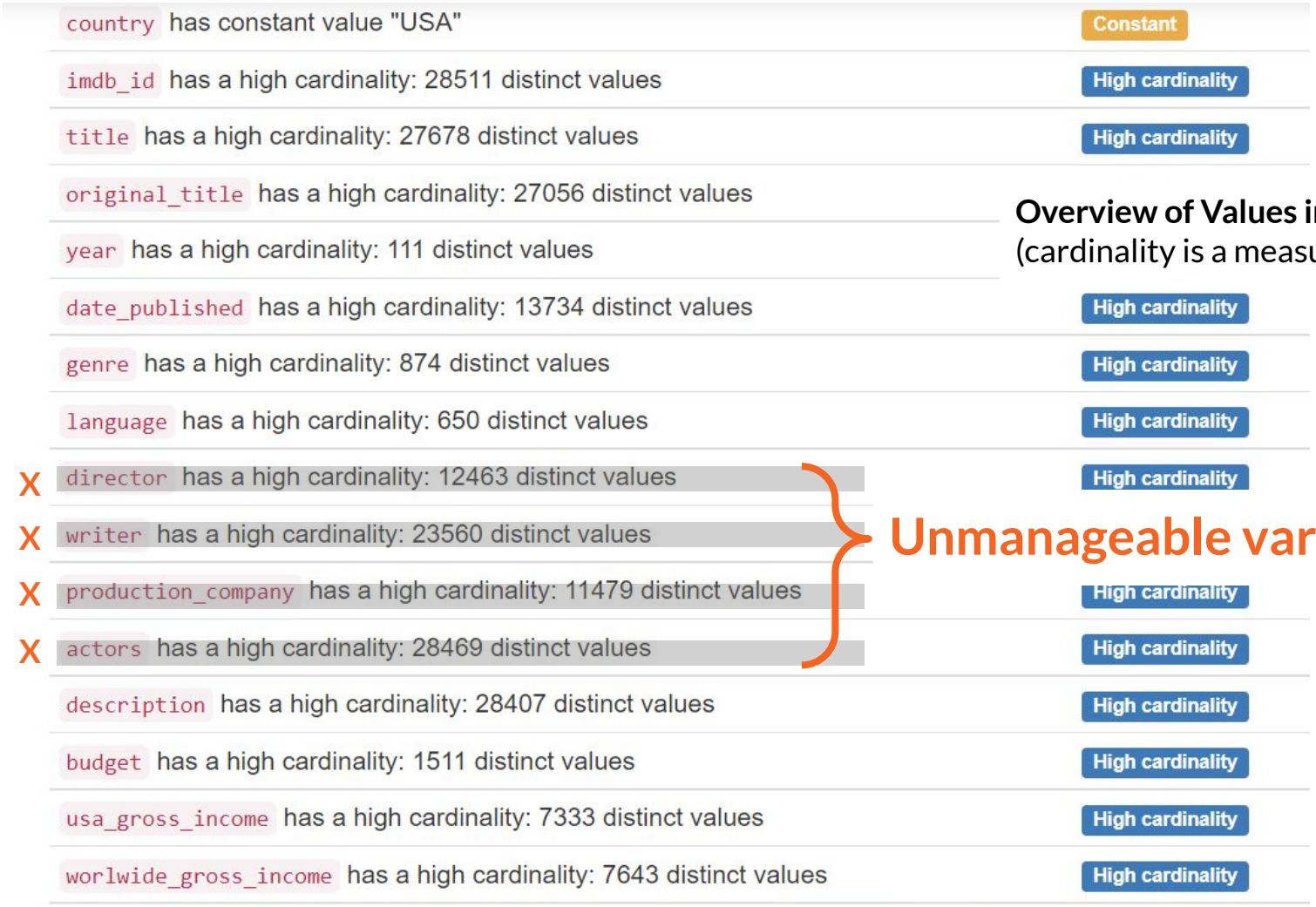
- **Internet Movies DataBase (IMDb)***
- **The Movies DataBase (TMDB)***
- **Film Awards (IMDb)* (not used)**

* See Appendix

Source: https://www.imdb.com
https://www.themoviedb.org/?language=en-US
https://www.kaggle.com/rounakbanik/the-movies-dataset

`country` has constant value "USA"	Constant
`imdb_id` has a high cardinality: 28511 distinct values	High cardinality
`title` has a high cardinality: 27678 distinct values	High cardinality
`original_title` has a high cardinality: 27056 distinct values	
`year` has a high cardinality: 111 distinct values	
`date_published` has a high cardinality: 13734 distinct values	High cardinality
`genre` has a high cardinality: 874 distinct values	High cardinality
`language` has a high cardinality: 650 distinct values	High cardinality
`director` has a high cardinality: 12463 distinct values	High cardinality
`writer` has a high cardinality: 23560 distinct values	High cardinality
`production_company` has a high cardinality: 11479 distinct values	High cardinality
`actors` has a high cardinality: 28469 distinct values	High cardinality
`description` has a high cardinality: 28407 distinct values	High cardinality
`budget` has a high cardinality: 1511 distinct values	High cardinality
`usa_gross_income` has a high cardinality: 7333 distinct values	High cardinality
`worlwide_gross_income` has a high cardinality: 7643 distinct values	High cardinality

Overview of Values in Each Variable
(cardinality is a measure of set size)



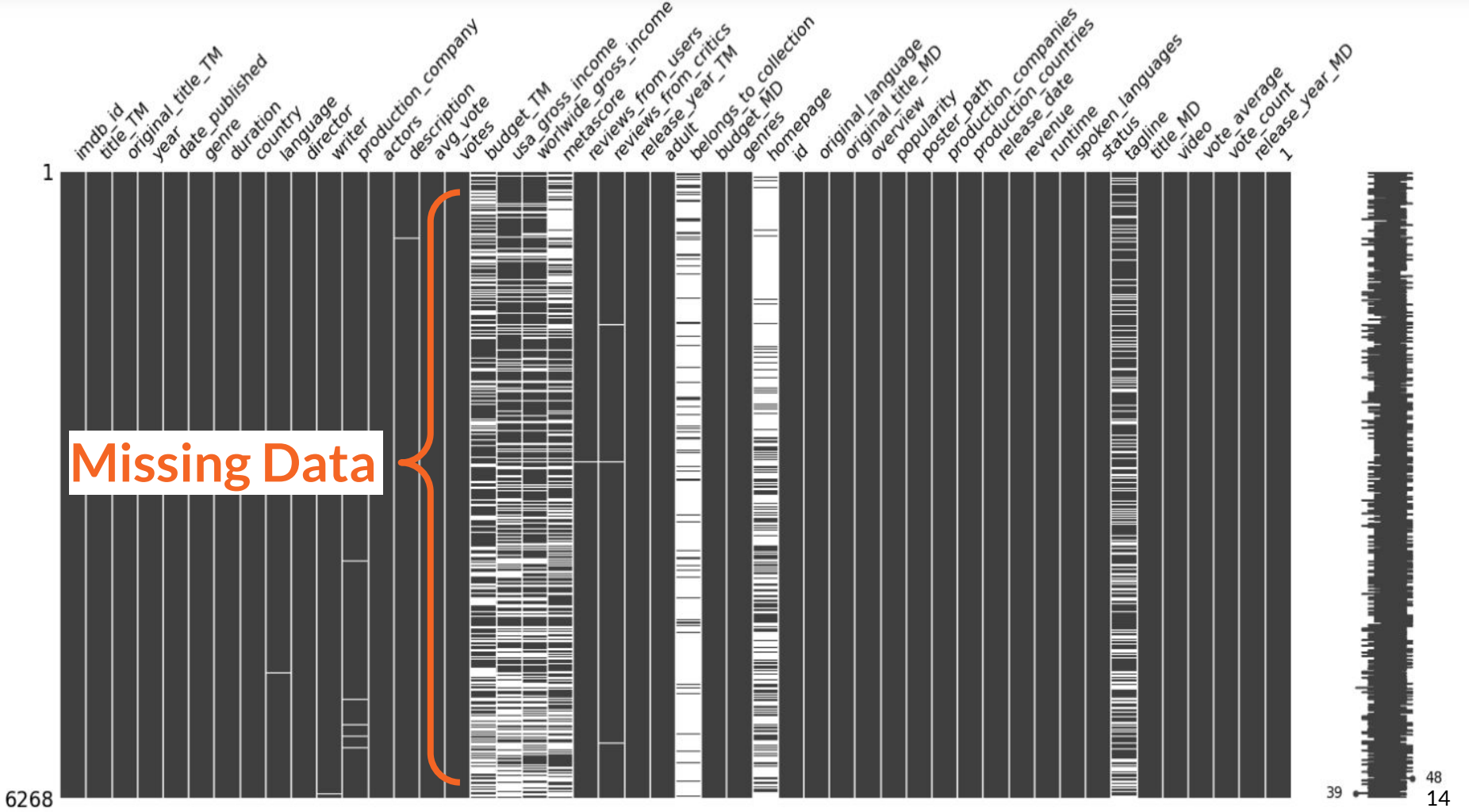
Overview of Values in Each Variable
(cardinality is a measure of set size)

Unmanageable variety

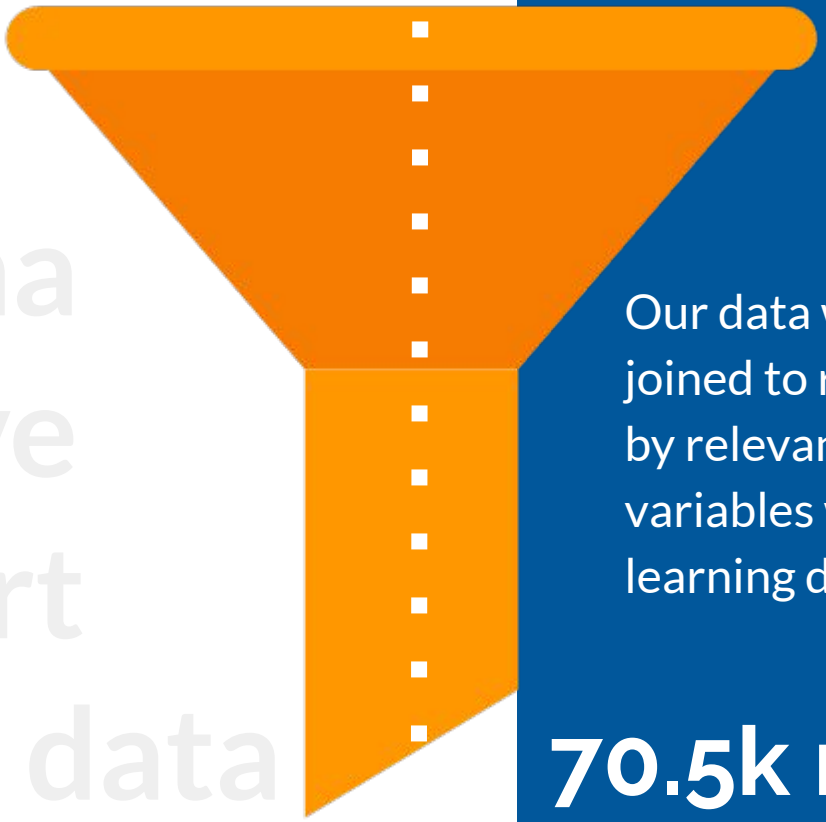
	`country` has constant value "USA"	Constant
	`imdb_id` has a high cardinality: 28511 distinct values	High cardinality
	`title` has a high cardinality: 27678 distinct values	High cardinality
	`original_title` has a high cardinality: 27056 distinct values	
	`year` has a high cardinality: 111 distinct values	
	`date_published` has a high cardinality: 13734 distinct values	High cardinality
	`genre` has a high cardinality: 874 distinct values	High cardinality
	`language` has a high cardinality: 650 distinct values	High cardinality
X	`director` has a high cardinality: 12463 distinct values	High cardinality
X	`writer` has a high cardinality: 23560 distinct values	High cardinality
X	`production_company` has a high cardinality: 11479 distinct values	High cardinality
X	`actors` has a high cardinality: 28469 distinct values	High cardinality
	`description` has a high cardinality: 28407 distinct values	High cardinality
X	`budget` has a high cardinality: 1511 distinct values	High cardinality
X	`usa_gross_income` has a high cardinality: 7333 distinct values	
X	`worlwide_gross_income` has a high cardinality: 7643 distinct values	High cardinality

Overview of Values in Each Variable
(cardinality is a measure of set size)

Low veracity (reliability)


imdb_id
title_TM
original_title_TM
year
date_published
genre
duration
country
language
director
writer
production_company
actors
description
avg_vote
votes
budget_TM
usa_gross_income
worlwide_gross_income
metascore
reviews_from_users
reviews_from_critics
release_year_TM
adult
belongs_to_collection
budget_MD
genres
homepage
id
original_language
original_title_MD
overview
popularity
poster_path
production_companies
production_countries
release_date
revenue
runtime
spoken_languages
status
tagline
title_MD
video
vote_average
vote_count
release_year_MD
1
1
6268
Missing Data
39
48

2M+ data entries



filter

drop na

remove

convert

model data

Our data was filtered, cleaned, and joined to remove missing data, filter by relevant content, and categorical variables were converted for machine learning data modeling.

70.5k rows of data

3 Total Variables Remaining

Technologies

Data Storage: PostgreSQL database 13.3 hosted on Amazon Web Services

* See Appendix

Technologies

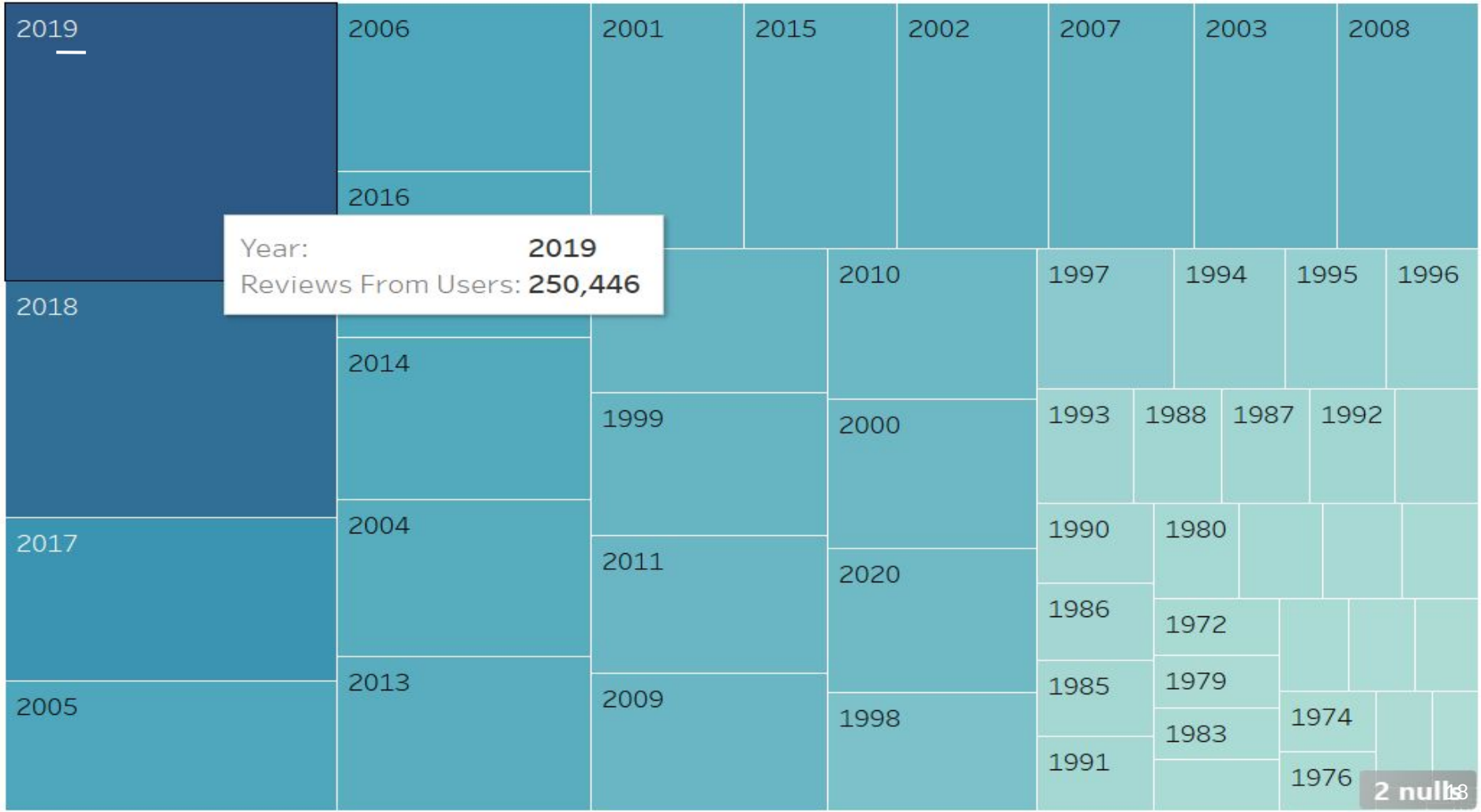
Data Cleaning + Analysis: Python 3.8.5, Pandas Library 1.3.2 (and dozens more*) in Jupyter Notebook 6.1.4

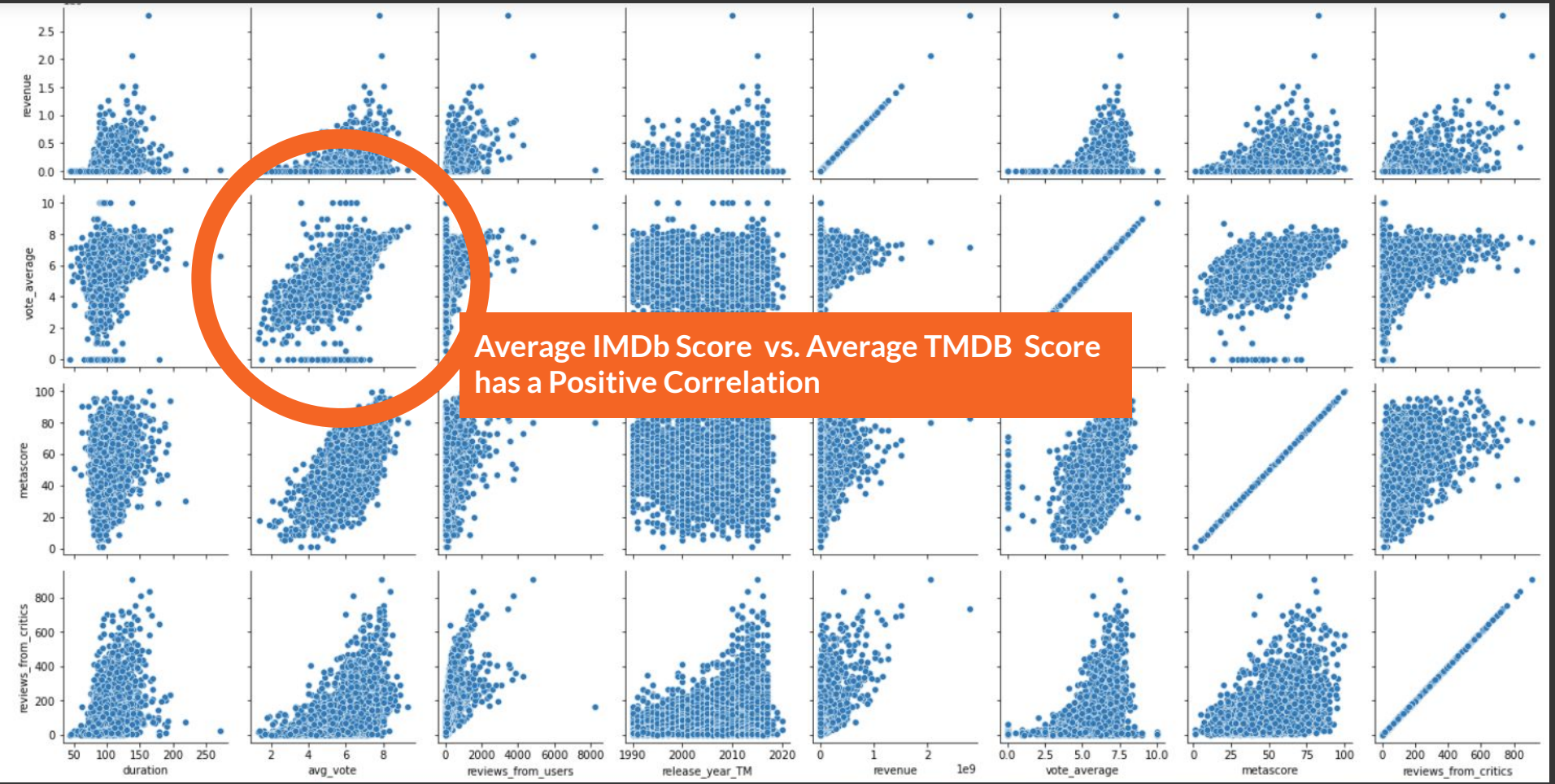
Data Storage: PostgreSQL database 13.3 hosted on Amazon Web Services

Machine Learning: Scikit-Learn 0.24 (+ dozens of additional Python libraries*) in Jupyter Notebook 6.1.4

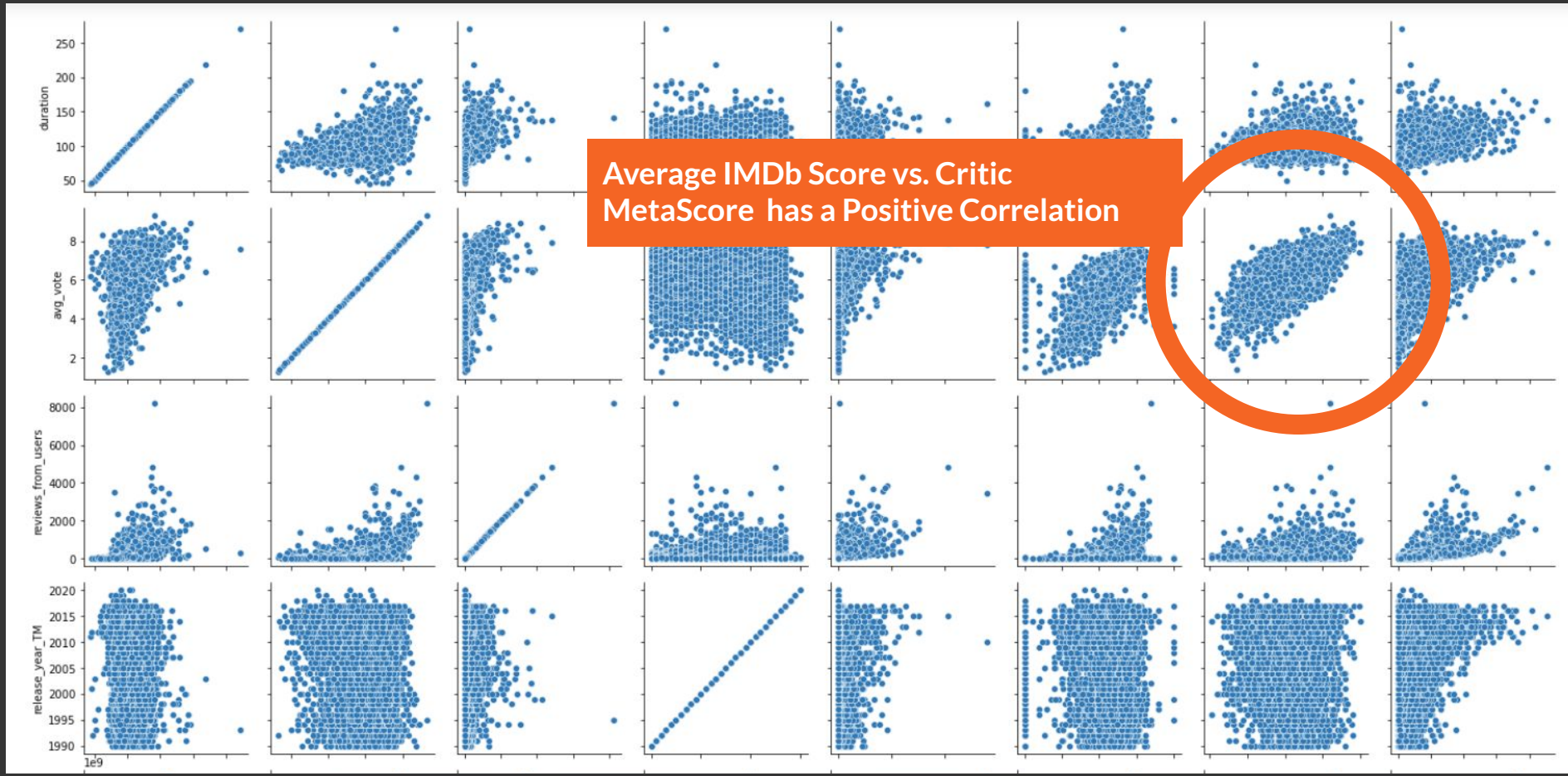
Dashboard: matplotlib 3.4.3 using Jupyter Notebook 6.1.4 + Tableau 2021.2.2

* See Appendix

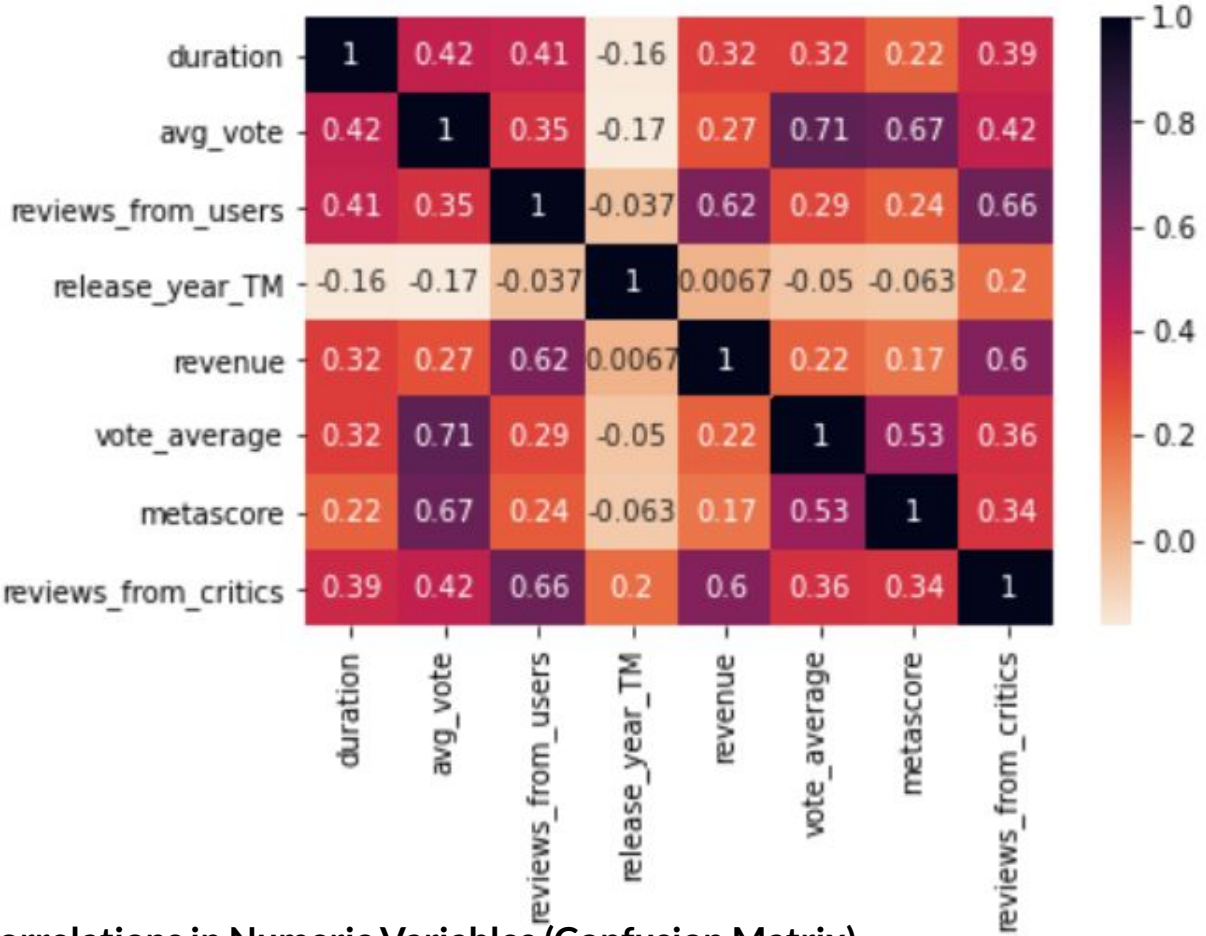

2019
2006
2001
2015
2002
2007
2003
2008
2016
Year: 2019
Reviews From Users: 250,446
2018
2010
1997
1994
1995
1996
2014
1999
2000
1993
1988
1987
1992
2004
2011
2020
1990
1980
1986
1972
2017
1985
1979
2013
2009
1998
1983
1974
2005
1991
1976
2 nulls



Relationships of Numerical Variables in Movies Data



Relationships of Numerical Variables in Movies Data



Correlations in Numeric Variables (Confusion Matrix)

CONCLUSION:

- Some positive correlation between revenue and user reviews
- Duration and Release Year has minimal correlations to other variables
- Note: Dropping revenue variable because data is incomplete

Some Data Issues We Encountered

1. Datasets were not well documented
2. Scraped raw data inconsistencies
3. Missing data in “success” variables
4. Lots of categorical data
5. Strings of data in JSON-like format
6. Proprietary or outdated data
7. Currency symbols created strings
8. Mismatched data types in columns
9. Multiple unique entries per cell
10. … and more.

Machine Learning Model

~~profits~~

ratings

~~awards~~

The ratings dataset contains:

- vote average (1-10)
- number of votes from IMDB users
- number of reviews from critics

Our Objective: Predict if a movie will be successful or not.

Model Comparison

~~profits~~

ratings

~~awards~~

	Accuracy	Precision	Recall	F1
Random Forest	0.803	0.390	0.240	0.290
Logistic Regression	0.830	0.570	0.090	0.150
Support Vector Machine	0.831	0.619	0.070	0.150
Deep Learning	0.831	0.580	0.090	0.150
Deep Learning Final	0.832	0.580	0.100	0.150

Our Objective: Predict if a movie will be successful or not.

Model Comparison

~~profits~~

ratings

~~awards~~

	Accuracy	Precision	Recall	F1
Random Forest	0.803	0.390	0.240	0.290
Logistic Regression	0.830	0.570	0.090	0.150
Support Vector Machine	0.831	0.619	0.070	0.150
Deep Learning	0.831	0.580	0.090	0.150
Deep Learning Final	0.832	0.580	0.100	0.150

Our Objective: Predict if a movie will be successful or not.

Model Comparison

~~profits~~

ratings

~~awards~~

	Accuracy	Precision	Recall	F1
Random Forest	0.803	0.390	0.240	0.290
Logistic Regression	0.830	0.570	0.090	0.150
Support Vector Machine	0.831	0.619	0.070	0.150
Deep Learning	0.831	0.580	0.090	0.150
Deep Learning Final	0.832	0.580	0.100	0.150

Our Objective: Predict if a movie will be successful or not.

Model Comparison

~~profits~~

ratings

~~awards~~

	Accuracy	Precision	Recall	F1
Random Forest	0.803	0.390	0.240	0.290
Logistic Regression	0.830	0.570	0.090	0.150
Support Vector Machine	0.831	0.619	0.070	0.150
Deep Learning	0.831	0.580	0.090	0.150
Deep Learning Final	0.832	0.580	0.100	0.150

Our Objective: Predict if a movie will be successful or not.

Model Comparison

~~profits~~

ratings

~~awards~~

	Accuracy	Precision	Recall	F1
Random Forest	0.803	0.390	0.240	0.290
Logistic Regression	0.830	0.570	0.090	0.150
Support Vector Machine	0.831	0.619	0.070	0.150
Deep Learning	0.831	0.580	0.090	0.150
Deep Learning Final	0.832	0.580	0.100	0.150

Our Objective: Predict if a movie will be successful or not.

Model Comparison

~~profits~~

ratings

~~awards~~

	Accuracy	Precision	Recall	F1
Random Forest	0.803	0.390	0.240	0.290
Logistic Regression	0.830	0.570	0.090	0.150
Support Vector Machine	0.831	0.619	0.070	0.150
Deep Learning	0.831	0.580	0.090	0.150
Deep Learning Final	0.832	0.580	0.100	0.150

Our Objective: Predict if a movie will be successful or not.

Model Comparison

~~profits~~

ratings

~~awards~~

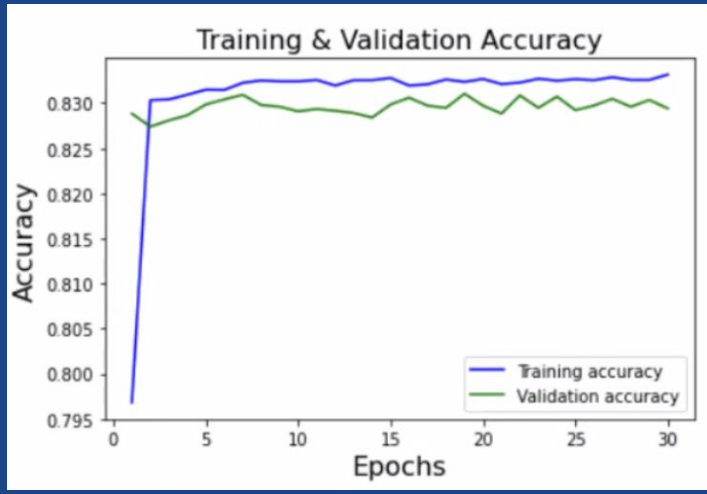
	Accuracy	Precision	Recall	F1
Random Forest	0.803	0.390	0.240	0.290
Logistic Regression	0.830	0.570	0.090	0.150
Support Vector Machine	0.831	0.619	0.070	0.150
Deep Learning	0.831	0.580	0.090	0.150
Deep Learning Final	0.832	0.580	0.100	0.150

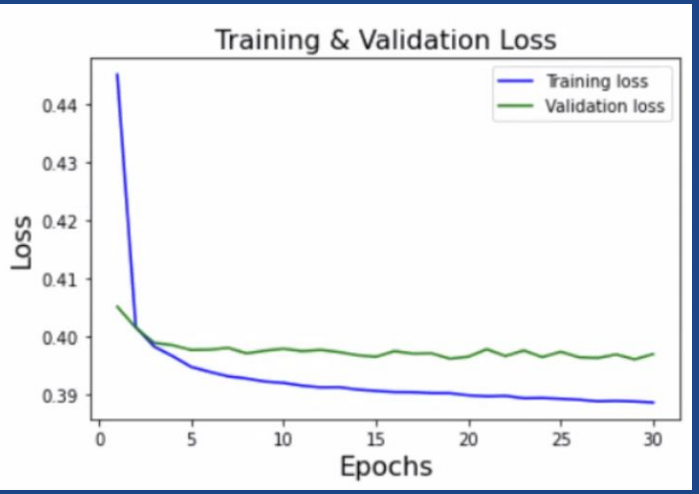
Our Objective: Predict if a movie will be successful or not.

—

random

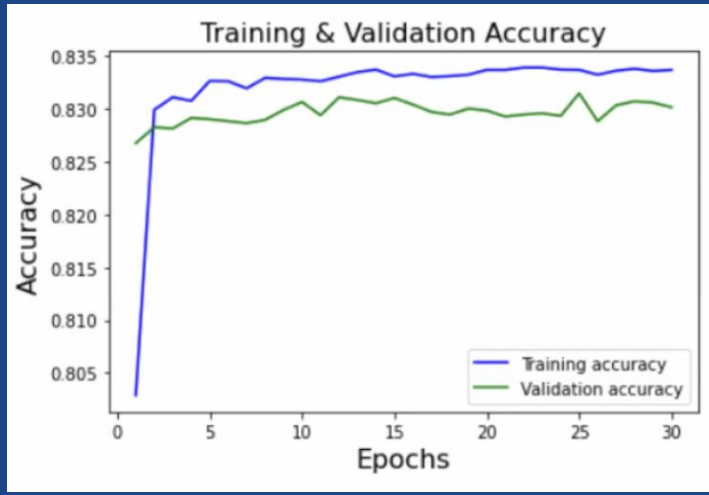
seed 67

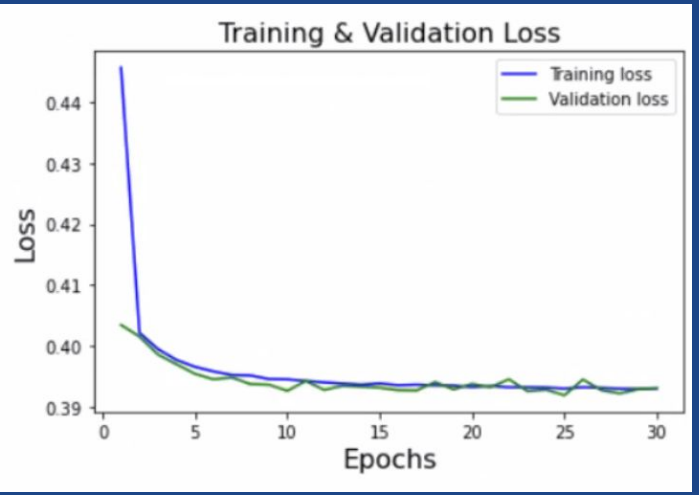




random

seed 189





The plot thickens...

Having a ton of data doesn't necessarily mean it is useful.

Where do we go from here?

- Seek Out Additional Data Sources (Awards, A-list, Rotten Tomato, etc.)
- Deep Dive Categorical Variables
- Natural Language Processing of Reviews
- Machine Learning Models for Predicting Future Success

Plot Twist?

What would we do differently? Maybe clone our team members to get after all the things left on the table!*

* We would definitely clone Kathy twice

:(

:(

:(

:(

:(

:)

Is it a coincidence that our instructor is the only happy camper in the bunch? Maybe...

fin

Appendix

ENCORE

Quantile statistics

Minimum	1.1
5-th percentile	3.1
Q1	4.8
median	5.8
Q3	6.5
95-th percentile	7.3
Maximum	9.7
Range	8.6
Interquartile range (IQR)	1.7

Descriptive statistics

Standard deviation	1.284809426
Coefficient of variation (CV)	0.2312437167
Kurtosis	0.05842961363
Mean	5.556083617
Median Absolute Deviation (MAD)	0.8
Skewness	-0.6237699936
Sum	158409.5
Variance	1.650735261
Monotonicity	Not monotonic

Average IMDb Score Summary Statistics

LIBRARIES AND TECHNOLOGIES LOADED ON OUR LOCAL MACHINE TO COMPLETE ETL

$ pip freeze
absl-py @ file:///C:/ci/absl-py_1623867338309/work
alabaster==0.7.12
anaconda-client==1.7.2
anaconda-project @ file:///tmp/build/80754af9/anaconda-project_1610472525955/work
anyio @ file:///C:/ci/anyio_1620153167783/work/dist
appdirs==1.4.4
argh==0.26.2
argon2-cffi @ file:///C:/ci/argon2-cffi_1613038019788/work
asn1crypto @ file:///tmp/build/80754af9/asn1crypto_1596577642040/work
astor==0.8.1
astroid @ file:///C:/ci/astroid_1613500971479/work
astropy @ file:///C:/ci/astropy_1617745665646/work
astunparse==1.6.3
async-generator==1.10
atomicwrites==1.4.0
attrs @ file:///tmp/build/80754af9/attrs_1604765588209/work
autopep8 @ file:///tmp/build/80754af9/autopep8_1615918855173/work
Babel @ file:///tmp/build/80754af9/babel_1607110387436/work
backcall==0.2.0
backports.shutil-get-terminal-size @ file:///tmp/build/80754af9/backports.shutil_get_terminal_size_1608222128777/work
bcrypt @ file:///C:/ci/bcrypt_1597918112552/work
beautifulsoup4 @ file:///home/linux1/recipes/ci/beautifulsoup4_1610988766420/work
bitarray @ file:///C:/ci/bitarray_1618435050316/work
bkcharts==0.2
black==19.10b0
bleach @ file:///tmp/build/80754af9/bleach_1612211392645/work
blinker==1.4
bokeh @ file:///C:/ci/bokeh_1620784051578/work
boto==2.49.0
Bottleneck==1.3.2
brotlipy==0.7.0
cachetools @ file:///tmp/build/80754af9/cachetools_1619597386817/work
certifi==2020.12.5
cffi @ file:///C:/ci/cffi_1613247308275/work
chardet @ file:///C:/ci/chardet_1607706910910/work
click @ file:///home/linux1/recipes/ci/click_1610990599742/work
cloudpickle @ file:///tmp/build/80754af9/cloudpickle_1598884132938/work
clyent==1.2.2
colorama==0.4.4
comtypes==1.1.9
contextlib2==0.6.0.post1
coverage @ file:///C:/ci/coverage_1614614910274/work
cryptography @ file:///C:/ci/cryptography_1616769432139/work
cycler==0.10.0
Cython @ file:///C:/ci/cython_1618435342622/work
cytoolz==0.11.0
dask @ file:///tmp/build/80754af9/dask-core_1617390489108/work
decorator==5.0.7
defusedxml @ file:///tmp/build/80754af9/defusedxml_1615228127516/work
diff-match-patch @ file:///tmp/build/80754af9/diff-match-patch_1594828741838/work
distributed @ file:///C:/ci/distributed_1617384292700/work
docutils @ file:///C:/ci/docutils_1617481620173/work
entrypoints==0.3
et-xmlfile==1.0.1
fastcache==1.1.0
filelock @ file:///home/linux1/recipes/ci/filelock_1610993975404/work
flake8 @ file:///tmp/build/80754af9/flake8_1615834841867/work
Flask @ file:///home/ktietz/src/ci/flask_1611932660458/work
fsspec @ file:///tmp/build/80754af9/fsspec_1617959894824/work
future==0.18.2
gast==0.3.3
gevent @ file:///C:/ci/gevent_1616773028237/work
glob2 @ file:///home/linux1/recipes/ci/glob2_1610991677669/work
google-auth @ file:///tmp/build/80754af9/google-auth_1600960338579/work
google-auth-oauthlib @ file:///tmp/build/80754af9/google-auth-oauthlib_1617120569401/work
google-pasta==0.2.0
greenlet @ file:///C:/ci/greenlet_1611958376725/work
grpcio @ file:///C:/ci/grpcio_1614884419385/work
h5py==2.10.0
HeapDict==1.0.1
html5lib @ file:///tmp/build/80754af9/html5lib_1593446221756/work
idna @ file:///home/linux1/recipes/ci/idna_1610986105248/work
imagecodecs @ file:///C:/ci/imagecodecs_1617996781001/work
imageio @ file:///tmp/build/80754af9/imageio_1617700267927/work
imagesize @ file:///Users/ktietz/demo/mc3/conda-bld/imagesize_1628863108022/work
imbalanced-learn @ file:///home/conda/feedstock_root/build_artifacts/imbalanced-learn_1613662486985/work
importlib-metadata @ file:///C:/ci/importlib-metadata_1617877486026/work
iniconfig @ file:///home/linux1/recipes/ci/iniconfig_1610983019677/work
intervaltree @ file:///tmp/build/80754af9/intervaltree_1598376443606/work
ipykernel==5.5.3
ipython==7.23.0
ipython-genutils==0.2.0
ipywidgets @ file:///tmp/build/80754af9/ipywidgets_1610481889018/work
isort @ file:///tmp/build/80754af9/isort_1616355431277/work
itsdangerous==1.1.0
jdcal==1.4.1
jedi==0.18.0
Jinja2 @ file:///tmp/build/80754af9/jinja2_1612213139570/work
joblib @ file:///tmp/build/80754af9/joblib_1613502643832/work
json5==0.9.5
jsonschema @ file:///tmp/build/80754af9/jsonschema_1602607155483/work
jupyter==1.0.0
jupyter-client==6.1.12
jupyter-console @ file:///tmp/build/80754af9/jupyter_console_1616615302928/work
jupyter-contrib-core==0.3.3
jupyter-core==4.7.1
jupyter-nbextensions-configurator @ file:///D:/bld/jupyter_nbextensions_configurator_1611341300533/work
jupyter-packaging @ file:///tmp/build/80754af9/jupyter-packaging_1613502826984/work
jupyter-server @ file:///C:/ci/jupyter_server_1616084265530/work
jupyterlab @ file:///tmp/build/80754af9/jupyterlab_1619133235951/work
jupyterlab-pygments @ file:///tmp/build/80754af9/jupyterlab_pygments_1601490720602/work
jupyterlab-server @ file:///tmp/build/80754af9/jupyterlab_server_1617134334258/work
jupyterlab-widgets @ file:///tmp/build/80754af9/jupyterlab_widgets_1609884341231/work
Keras @ file:///tmp/build/80754af9/keras_split_1593112142734/work
Keras-Applications @ file:///tmp/build/80754af9/keras-applications_1594366238411/work
Keras-Preprocessing @ file:///tmp/build/80754af9/keras-preprocessing_1612283640596/work
keyring @ file:///C:/ci/keyring_1614630298708/work
kiwisolver @ file:///C:/ci/kiwisolver_1612282618948/work
lazy-object-proxy @ file:///C:/ci/lazy-object-proxy_1616529290879/work
libarchive-c @ file:///tmp/build/80754af9/python-libarchive-c_1617780486945/work
llvmlite==0.36.0
locket==0.2.1
lxml @ file:///C:/ci/lxml_1616443391272/work
Markdown @ file:///C:/ci/markdown_1614364005059/work
MarkupSafe @ file:///C:/ci/markupsafe_1594405949945/work
matplotlib @ file:///C:/ci/matplotlib-suite_1613408055530/work
matplotlib-inline==0.1.2
mccabe==0.6.1
menuinst==1.4.16
mistune @ file:///C:/ci/mistune_1594373272338/work
mkl-fft==1.3.0
mkl-random @ file:///C:/ci/mkl_random_1618854593605/work
mkl-service==2.3.0
mock @ file:///tmp/build/80754af9/mock_1607622725907/work
more-itertools @ file:///tmp/build/80754af9/more-itertools_1613676688952/work
mpmath==1.2.1
msgpack @ file:///C:/ci/msgpack-python_1612287191162/work
multipledispatch==0.6.0
mypy-extensions==0.4.3
nbclassic @ file:///tmp/build/80754af9/nbclassic_1616085367084/work
nbclient @ file:///tmp/build/80754af9/nbclient_1614364831625/work
nbconvert @ file:///C:/ci/nbconvert_1601914921407/work
nbformat @ file:///tmp/build/80754af9/nbformat_1617383369282/work
nest-asyncio @ file:///tmp/build/80754af9/nest-asyncio_1613680548246/work
networkx @ file:///tmp/build/80754af9/networkx_1598376031484/work
nltk @ file:///tmp/build/80754af9/nltk_1618327084230/work
nose @ file:///tmp/build/80754af9/nose_1606773131901/work
notebook @ file:///C:/ci/notebook_1616443616158/work
numba @ file:///C:/ci/numba_1616774290339/work
numexpr @ file:///C:/ci/numexpr_1618856761305/work
numpy @ file:///C:/ci/numpy_and_numpy_base_1618497408168/work
numpydoc @ file:///tmp/build/80754af9/numpydoc_1605117425582/work
oauthlib @ file:///tmp/build/80754af9/oauthlib_1623060228408/work
olefile==0.46
openpyxl @ file:///tmp/build/80754af9/openpyxl_1615411699337/work
opt-einsum @ file:///tmp/build/80754af9/opt_einsum_1621500238896/work
packaging @ file:///tmp/build/80754af9/packaging_1611952188834/work
pandas @ file:///C:/ci/pandas_1618365631664/work
pandocfilters @ file:///C:/ci/pandocfilters_1605102427207/work
paramiko @ file:///tmp/build/80754af9/paramiko_1598886428689/work
parso==0.8.2
partd @ file:///tmp/build/80754af9/partd_1618000087440/work
path @ file:///C:/ci/path_1614022434895/work
pathlib2 @ file:///C:/ci/pathlib2_1607025068091/work
pathspec==0.7.0

LIBRARIES AND TECHNOLOGIES LOADED ON OUR LOCAL MACHINE TO COMPLETE ETL (cont.)

athlib2 @ file:///C:/ci/pathlib2_1607025068091/work
pathspec==0.7.0
patsy==0.5.1
pep8==1.7.1
pexpect @ file:///tmp/build/80754af9/pexpect_1605563209008/work
pickleshare==0.7.5
Pillow @ file:///C:/ci/pillow_1617386319301/work
pkginfo==1.7.0
pluggy @ file:///C:/ci/pluggy_1615976530170/work
ply==3.11
prometheus-client @ file:///tmp/build/80754af9/prometheus_client_1618088486455/work
prompt-toolkit==3.0.18
protobuf==3.17.2
psutil @ file:///C:/ci/psutil_1612298033174/work
psycopg2 @ file:///C:/ci/psycopg2_1612298715048/work
ptyprocess @ file:///tmp/build/80754af9/ptyprocess_1609355006118/work/dist/ptyprocess-0.7.0-py2.py3-none-any.whl
py @ file:///tmp/build/80754af9/py_1607971587848/work
pyasn1==0.4.8
pyasn1-modules==0.2.8
pycodestyle @ file:///home/ktietz/src/ci_mi/pycodestyle_1612807597675/work
pycosat==0.6.3
pycparser @ file:///tmp/build/80754af9/pycparser_1594388511720/work
pycrypto==2.6.1
pycurl==7.43.0.6
pydocstyle @ file:///tmp/build/80754af9/pydocstyle_1616182067796/work
pyerfa @ file:///C:/ci/pyerfa_1619391058121/work
pyflakes @ file:///home/ktietz/src/ci_ipy2/pyflakes_1612551159640/work
Pygments==2.8.1
PyJWT @ file:///C:/ci/pyjwt_1619651810943/work
pylint @ file:///C:/ci/pylint_1617135975189/work
pyls-black @ file:///tmp/build/80754af9/pyls-black_1607553132291/work
pyls-spyder @ file:///tmp/build/80754af9/pyls-spyder_1613849700860/work
PyNaCl @ file:///C:/ci/pynacl_1595009241355/work
pyodbc===4.0.0-unsupported
pyOpenSSL @ file:///tmp/build/80754af9/pyopenssl_1608057966937/work
pyparsing @ file:///home/linux1/recipes/ci/pyparsing_1610983426697/work
pyreadline==2.1
pyrsistent @ file:///C:/ci/pyrsistent_1600123688363/work
PySocks @ file:///C:/ci/pysocks_1594394709107/work
pytest==6.2.3
python-dateutil==2.8.1
python-jsonrpc-server @ file:///tmp/build/80754af9/python-jsonrpc-server_1600278539111/work
python-language-server @ file:///tmp/build/80754af9/python-language-server_1607972495879/work
pytz @ file:///tmp/build/80754af9/pytz_1612215392582/work
PyWavelets @ file:///C:/ci/pywavelets_1601658407053/work
pywin32==300
pywin32-ctypes @ file:///C:/ci/pywin32-ctypes_1594392691209/work
pywinpty==0.5.7
PyYAML==5.4.1
pyzmq==22.0.3
QDarkStyle==2.8.1
QtAwesome @ file:///tmp/build/80754af9/qtawesome_1615991616277/work
qtconsole @ file:///tmp/build/80754af9/qtconsole_1616775094278/work
QtPy==1.9.0
regex @ file:///C:/ci/regex_1617569892025/work
requests @ file:///tmp/build/80754af9/requests_1608241421344/work
requests-oauthlib==1.3.0
rope @ file:///tmp/build/80754af9/rope_1602264064449/work
rsa @ file:///tmp/build/80754af9/rsa_1614366226499/work
Rtree @ file:///C:/ci/rtree_1618421019532/work
ruamel-yaml-conda @ file:///C:/ci/ruamel_yaml_1616016865685/work
scikit-image==0.18.1
scikit-learn @ file:///C:/ci/scikit-learn_1614446716349/work
scipy @ file:///C:/ci/scipy_1618856134946/work
seaborn @ file:///tmp/build/80754af9/seaborn_1608578541026/work
Send2Trash @ file:///tmp/build/80754af9/send2trash_1607525499227/work
simplegeneric==0.8.1
singledispatch @ file:///tmp/build/80754af9/singledispatch_1614366001199/work
six==1.15.0
sniffio @ file:///C:/ci/sniffio_1614030522573/work
snowballstemmer @ file:///tmp/build/80754af9/snowballstemmer_1611258885636/work
sortedcollections @ file:///tmp/build/80754af9/sortedcollections_1611172717284/work
sortedcontainers @ file:///tmp/build/80754af9/sortedcontainers_1606865132123/work
soupsieve @ file:///tmp/build/80754af9/soupsieve_1616183228191/work
Sphinx @ file:///tmp/build/80754af9/sphinx_1620777493457/work
sphinxcontrib-applehelp @ file:///home/ktietz/src/ci/sphinxcontrib-applehelp_1611920841464/work
sphinxcontrib-devhelp @ file:///home/ktietz/src/ci/sphinxcontrib-devhelp_1611920923094/work
sphinxcontrib-htmlhelp @ file:///home/ktietz/src/ci/sphinxcontrib-htmlhelp_1611920974801/work
sphinxcontrib-jsmath @ file:///home/ktietz/src/ci/sphinxcontrib-jsmath_1611920942228/work
sphinxcontrib-qthelp @ file:///home/ktietz/src/ci/sphinxcontrib-qthelp_1611921055322/work
sphinxcontrib-serializinghtml @ file:///home/ktietz/src/ci/sphinxcontrib-serializinghtml_1611920755253/work
sphinxcontrib-websupport @ file:///tmp/build/80754af9/sphinxcontrib-websupport_1597081412696/work
spyder @ file:///C:/ci/spyder_1616776686228/work
spyder-kernels @ file:///C:/ci/spyder-kernels_1614030834721/work
SQLAlchemy @ file:///C:/ci/sqlalchemy_1618090063585/work
statsmodels==0.12.2
sympy @ file:///C:/ci/sympy_1618255481827/work
tables==3.6.1
tblib @ file:///tmp/build/80754af9/tblib_1597928476713/work
tensorboard @ file:///home/builder/ktietz/aggregate/tensorflow_recipes/ci_te/tensorboard_1614593728657/work/tmp_pip_dir
tensorboard-plugin-wit==1.6.0
tensorflow==2.3.0
tensorflow-estimator @ file:///home/builder/ktietz/aggregate/tensorflow_recipes/ci_baze37/tensorflow-estimator_1622026529081/work/tensorflow_estimator-2.5.0-py2.py3-none-any.whl
termcolor==1.1.0
terminado==0.9.4
testpath @ file:///home/ktietz/src/ci/testpath_1611930608132/work
textdistance @ file:///tmp/build/80754af9/textdistance_1612461398012/work
threadpoolctl @ file:///tmp/tmp9twdgx9k/threadpoolctl-2.1.0-py3-none-any.whl
three-merge @ file:///tmp/build/80754af9/three-merge_1607553261110/work
tifffile @ file:///tmp/build/80754af9/tifffile_1619636090847/work
toml @ file:///tmp/build/80754af9/toml_1616166611790/work
toolz @ file:///home/linux1/recipes/ci/toolz_1610987900194/work
tornado==6.1
tqdm @ file:///tmp/build/80754af9/tqdm_1615925068909/work
traitlets==5.0.5
typed-ast @ file:///C:/ci/typed-ast_1610484654578/work
typing-extensions @ file:///home/ktietz/src/ci_mi/typing_extensions_1612808209620/work
ujson @ file:///C:/ci/ujson_1611244941645/work
unicodecsv==0.14.1
urllib3 @ file:///tmp/build/80754af9/urllib3_1615837158687/work
watchdog @ file:///C:/ci/watchdog_1612471244702/work
wcwidth==0.2.5
webencodings==0.5.1
Werkzeug @ file:///home/ktietz/src/ci/werkzeug_1611932622770/work
widgetsnbextension==3.5.1
win-inet-pton @ file:///C:/ci/win_inet_pton_1605306165655/work
win-unicode-console==0.5
wincertstore==0.2
wrapt==1.12.1
xlrd @ file:///tmp/build/80754af9/xlrd_1608072521494/work
XlsxWriter @ file:///tmp/build/80754af9/xlsxwriter_1617224712951/work
xlwings==0.23.0
xlwt==1.3.0
yapf @ file:///tmp/build/80754af9/yapf_1615749224965/work
zict==2.0.0
zipp @ file:///tmp/build/80754af9/zipp_1615904174917/work
zope.event==4.5.0
zope.interface @ file:///C:/ci/zope.interface_1616357230604/work
(py37nbext)

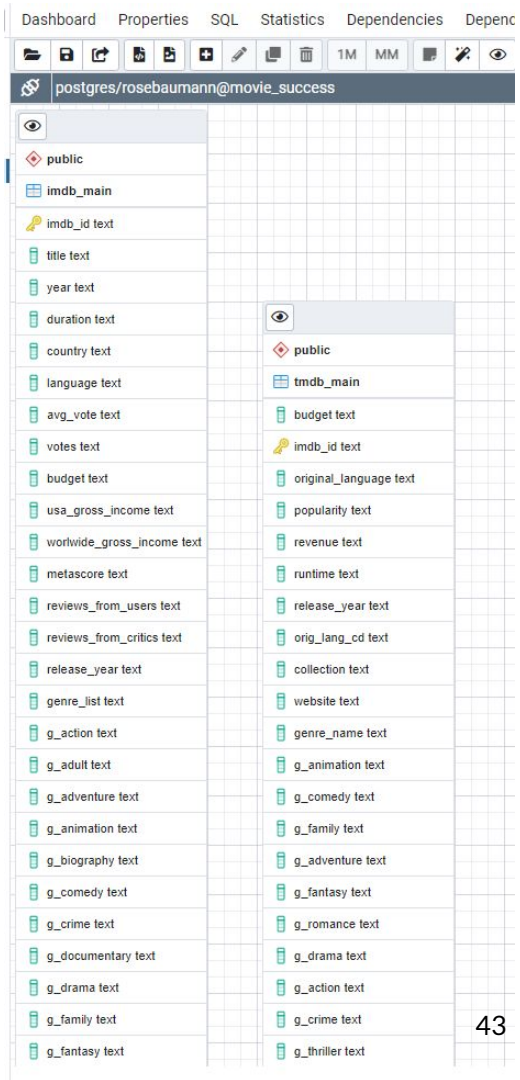
Our Data Exploration

Data >
AWS RD + S3 >
Postgres >
Jupyter Notebook >
Pandas >

55 Categorical Variables
13 Numerical Variables
\+ 2 Boolean Variables

70 Total Variables

Our ERD


Dashboard
Properties
SQL
Statistics
Dependencies
Depenc
1M
MM
postgres/rosebaumann@movie_success
public
imdb_main
imdb_id text
title text
year text
duration text
country text
language text
avg_vote text
votes text
budget text
usa_gross_income text
worlwide_gross_income text
metascore text
reviews_from_users text
reviews_from_critics text
release_year text
genre_list text
g_action text
g_adult text
g_adventure text
g_animation text
g_biography text
g_comedy text
g_crime text
g_documentary text
g_drama text
g_family text
g_fantasy text
public
tmdb_main
budget text
imdb_id text
original_language text
popularity text
revenue text
runtime text
release_year text
orig_lang_cd text
collection text
website text
genre_name text
g_animation text
g_comedy text
g_family text
g_adventure text
g_fantasy text
g_romance text
g_drama text
g_action text
g_crime text
g_thriller text

Data Sources

1 - source: IMDb Movies Extensive Dataset

https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset

file: https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset/download

Contains metadata scraped from IMDB movies with at least 100 votes as of 1/1/2020

2 - The Movies Dataset (TMDB)

source: https://www.kaggle.com/rounakbanik/the-movies-dataset?select=movies_metadata.csv

file: https://www.kaggle.com/rounakbanik/the-movies-dataset/download

Contains metadata from the Full MovieLens Dataset for movies released on or before July 2017.

3 - Film Awards (IMDB)

source: https://www.kaggle.com/iwooloowi/film-awards-imdb

Last updated 3/25/2020

—

- Link to Repository
- Link to ReadMe.md file

Link to Profile Report