**INTRODUCTION**

* 1. **PYTHON :**
  2. **About Python:**
* Python is a high-level, general-purpose, open source, strictly typed programming language. The language provides constructs intended to enable clear programs on both a small and large scale.
* Python was created By Guido van Rossum.
* The Python Software Foundation (PSF) is the organization behind Python.
* Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.
* Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

**Python versions:**

* + - First released in1991.
    - Python 2.0 was released on 16 October2000
    - Python 3.0 was released on 3 December2008

**Current Versions:**

* + - 3.6.3
    - 2.7.14

**Python features:**

Some of the features of python include :-

* + - Easy to understand
* GUI Programming Support
  + - Dynamic
    - Portable Language
    - Integrated and Interpreted
    - Large Standard Library
    - Object oriented
    - Multipurpose
    - Strongly typed
    - Open Source

**Python is mainly used in many domains:**

* + - Web Development
    - Data Analysis
    - Machine Learning
    - Internet Of Things
    - GUI Development
    - Image processing
    - Data visualization
    - Game Development

**IDLE:**

IDLE is an integrated development environment for Python, which has been bundled with the default implementation of the language.

IDLE can be used to execute a single statement just like Python Shell and also to create, modify and execute Python scripts. IDLE provides a fully-featured text editor to create Python scripts that includes features like syntax highlighting, auto completion and smart indent. It also has a debugger with stepping and breakpoints features.

**Anaconda**

Anaconda is a open source Distribution for data science and machine learning using python. It includes hundreds of popular data science packages and the conda package and virtual environment manager for Windows, Linux, and MacOS. Conda makes it quick and easy to install, run, and upgrade complex data science and machine learning environments like scikit-learn, TensorFlow, and SciPy. Anaconda Distribution is the foundation of millions of data science projects as well as Amazon Web Service Machine Learning AMIs and Anaconda for Microsoft on Azure and Windows.

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

* Jupyter Lab
* Jupyter Notebook
* QtConsole
* Spyder
* Glue
* Orange
* RStudio
* Visual Studio Code
  1. **Packages**

A package is a hierarchical file directory structure that defines a single Python application environment that consists of modules and sub-packages and sub-sub-packages, and so on.

* + 1. **NumPy**

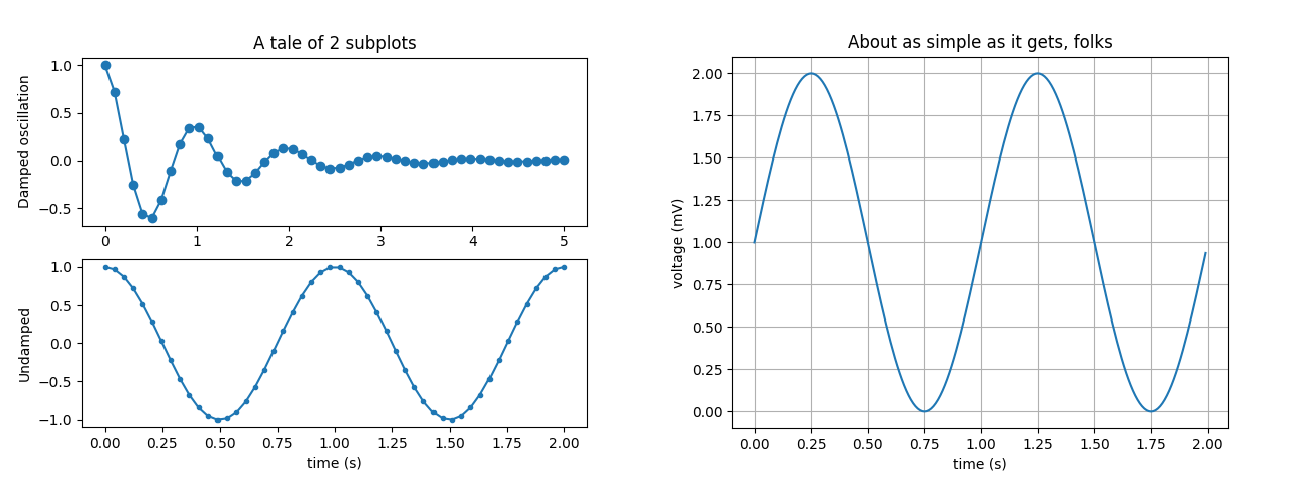
NumPy is a python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

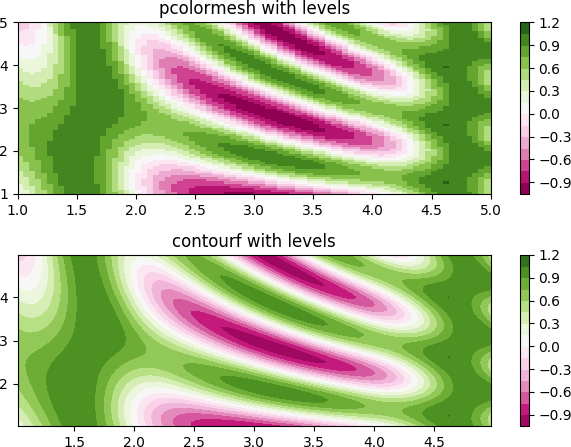
NumPy is the fundamental package for scientific computing with Python. It contains among other things:

* a powerful N-dimensional array object
* sophisticated (broadcasting)functions
* tools for integrating C/C++ and Fortran code
* useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPyto seamlessly and speedily integrate with a wide variety of databases.

* + 1. **Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, the jupyter notebook, web application servers, and four graphical user interface toolkits.



Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatterplots, etc., with just a few lines of code.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly whencombined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

* + 1. **Scikit-learn**

Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

Scikit-learn provides machine learning libraries for python. Some of the features of Scikit-learn includes:

* + - * Simple and efficient tools for data mining and dat aanalysis
      * Accessible to everybody, and reusable in various contexts
      * Built on NumPy, SciPy, and matplotlib
      * Open source, commercially usable – BSD license
    1. **Pandas**

Pandas is an open source, BSD-licensed library providing high- performance, easy-to-use data structures and data analysis tools for the Python programming language.

Pandas library is well suited for data manipulation and analysis using python. In particular, it offers data structures and operations for manipulating numerical tables

and time series.

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

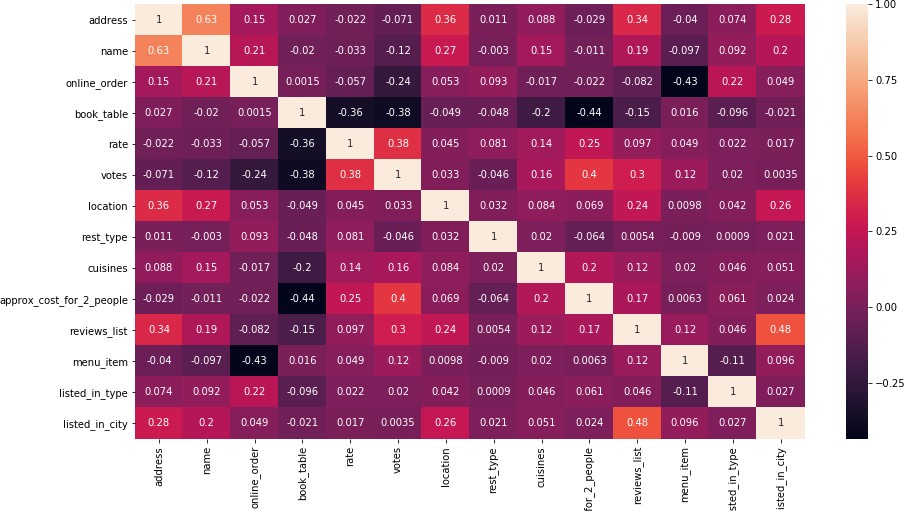
* + 1. **Seaborn**

Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics. Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures. Here is some of the functionality that seaborn offers:

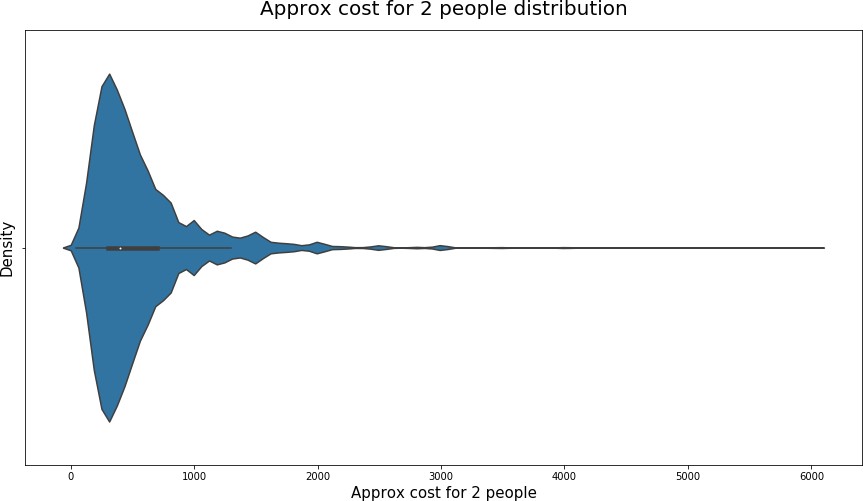
* A dataset-oriented API for examining relationships between multiple variables
* Specialized support for using categorical variables to show observations or aggregate statistics
* Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data
* Automatic estimation and plotting of linear regression models for different kinds dependent variables
* Convenient views onto the overall structure of complex datasets.

Seaborn aims to make visualization a central part of exploring and understanding data. Its dataset-oriented plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots

**Heatmap:**



**Violinplot**

****

**TRAINING WORKUNDERTAKEN**

**Collecting Data From Kaggle**

Kaggle is a platform for predictive modelling and analytics competitions in which statisticians and dataminers compete to produce the best models for predicting and describing the datasets uploaded by companies and users. This crowd sourcing approach relies on the fact that there are countless strategies that can be applied to any predictive modelling task and it is impossible to know beforehand which technique or analyst will be most effective. On 8 March 2017, Google announced that they were acquiring Kaggle. They will join the Google Cloud team and continue to be a distinct brand. In January 2018, Booz Allen and Kaggle launched Data Science Bowl, a machine learning competition to analyze cell images and identify nuclei.

**For Movie Recommendation System Data Set:**

**Context:**

IMDb is the most popular movie website and it combines movie plot description, Metasore ratings, critic and user ratings and reviews, release dates, and many more aspects.

The website is well known for storing almost every movie that has ever been released (the oldest is from 1874 - "Passage de Venus") or just planned to be released (newest movie is from 2027 - "Avatar 5").

IMDb stores information related to more than 6 million titles (of which almost 500,000 are featured films) and it is owned by Amazon since 1998.

**Content:**

The movies dataset includes 81,273 movies with attributes such as movie description, average rating, number of votes, genre, etc.

The ratings dataset includes 81,273 rating details from demographic perspective.

The names dataset includes 175,719 cast members with personal attributes such as birth details, death details, height, spouses, children, etc.

The title principals dataset includes 377,848 cast members roles in movies with attributes such as IMDb title id, IMDb name id, order of importance in the movie, role, and characters played.

**DATASCIENCE**

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data in various forms, both structured and unstructured, similar to data mining. Data science is a "concept to unify statistics, data analysis, machine learning and their related methods" in order to "understand and analyze actual phenomena" with data. It employs techniques and theories drawn from many fields within the context of mathematics, statistics, information science, and computer science.

Turing award winner JiGray imagined data science as a "fourth paradigm" of science (empirical, theoretical, computational and now data-driven) and asserted that "everything about science is changing because of the impact of information technology "and the data deluge. When Harvard Business Review called it "The Sexiest Job of the 21st Century" the term became a buzzword, and is now often applied to business analytics, business intelligence, predictive modeling, or any arbitrary use of data, or used as a glamorized term for statistics. In many cases, earlier approaches and solutions are now simply rebranded as "data science" to be more attractive, which can cause the term to become "dilute[d] beyond usefulness." While many university programs now offer a data science degree, there exists no consensus on a definition or suitable curriculum contents. Because of the current popularity of this term, there are many "advocacy efforts" surrounding the field. To its discredit, however, many data science and big data projects fail to deliver useful results, often as a result of poor management and utilization of resources.

**SOURCE CODE AND OUTPUT:**

* Importing Packages:

|  |
| --- |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  from sklearn.model\_selection import train\_test\_split  from sklearn.linear\_model import LogisticRegression  from sklearn.metrics import accuracy\_score  from sklearn.preprocessing import LabelEncoder  from sklearn.metrics import classification\_report  from sklearn.metrics import confusion\_matrix  from sklearn.metrics import r2\_score  import seaborn as sns  import warnings  warnings.filterwarnings("ignore")  le = LabelEncoder() |

* Load the dataset:

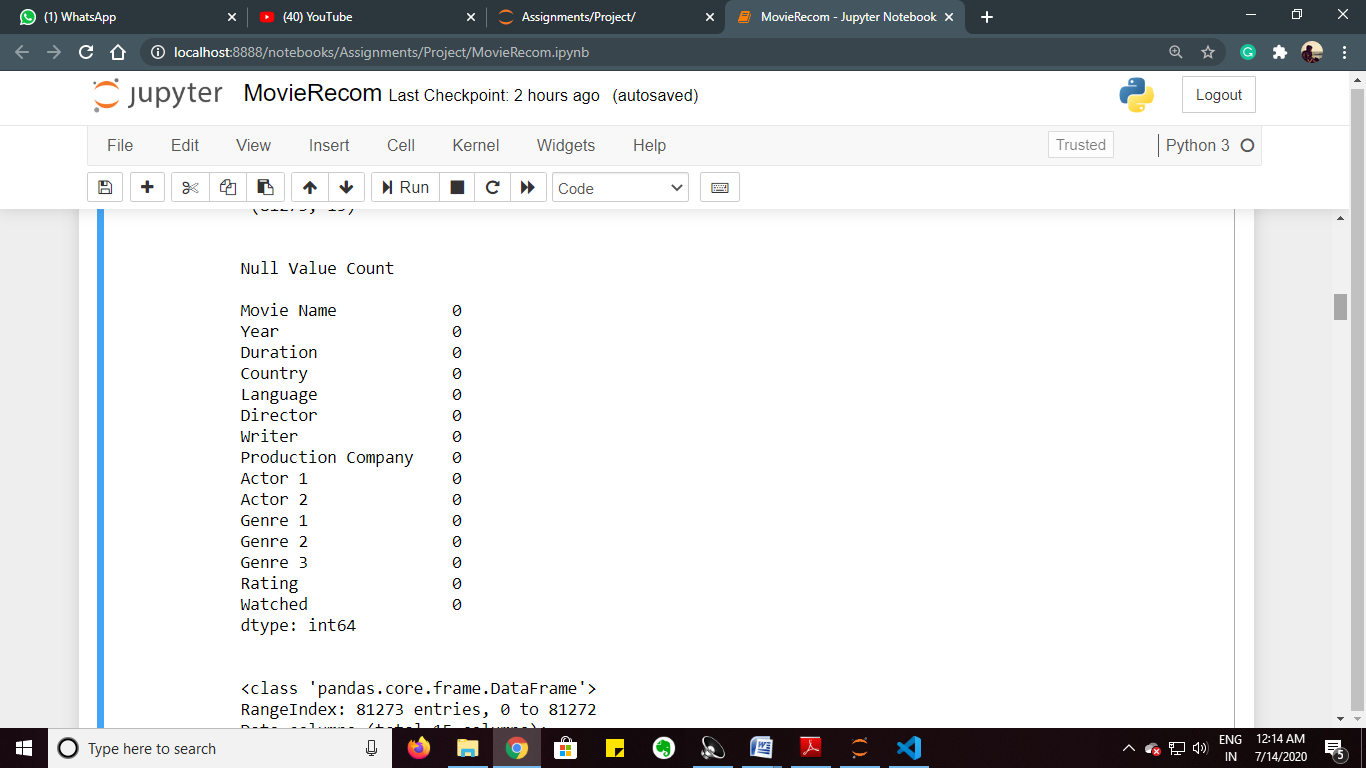
|  |
| --- |
| url='movies.csv'  dataset=pd.read\_csv(url)  df=pd.read\_csv(url)  dataset['Country']=dataset['Country'].astype(str)  dataset['Actor 1']=dataset['Actor 1'].astype(str)  dataset['Actor 2']=dataset['Actor 2'].astype(str) |



**DATA PRE-PROCESSING**

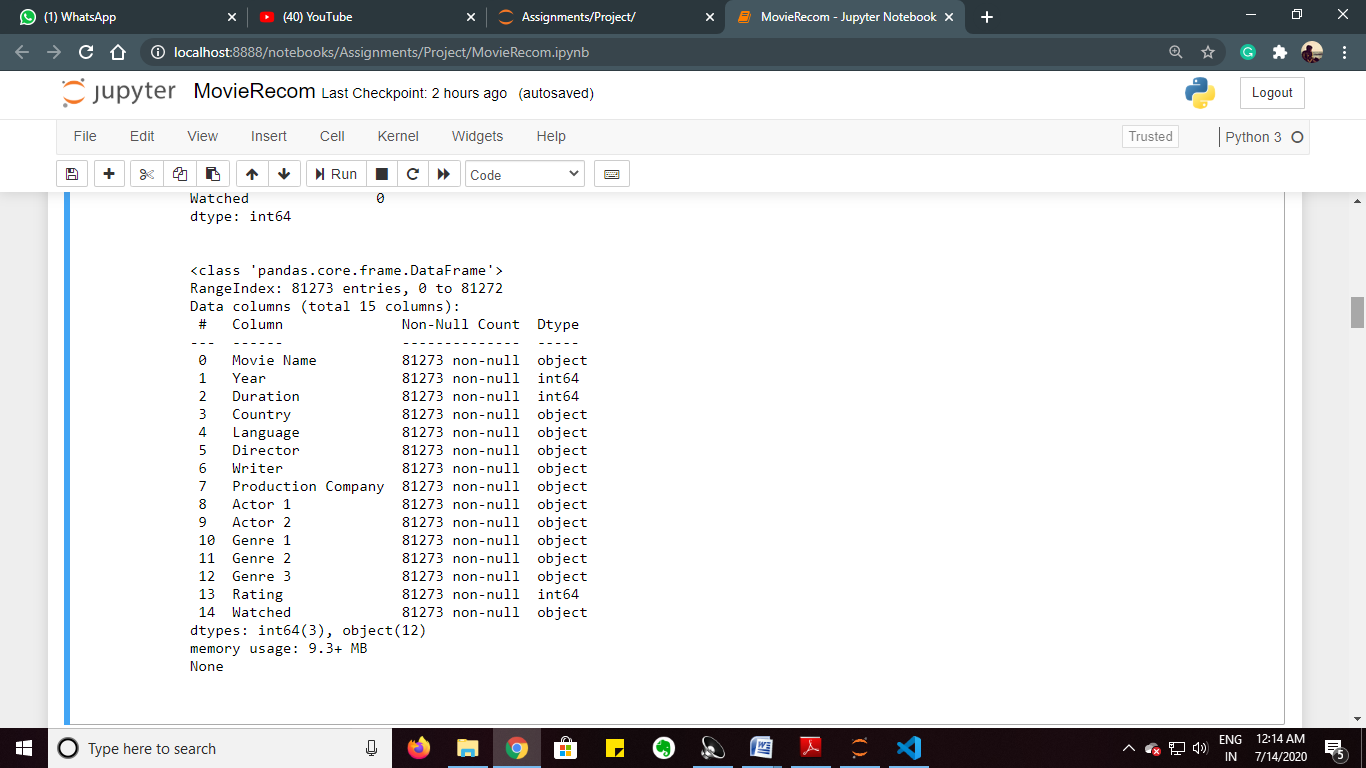
* **Counting NULL values for different columns**

|  |
| --- |
| dataset['Country']=dataset['Country'].astype(str)  dataset['Actor 1']=dataset['Actor 1'].astype(str)  dataset['Actor 2']=dataset['Actor 2'].astype(str)  print(dataset.head())  print(dataset.shape)  print('\n')  print('Null Value Count\n')  print(dataset.isnull().sum()) |

****

* **Information about dataset:**

|  |
| --- |
| print(dataset.info()) |

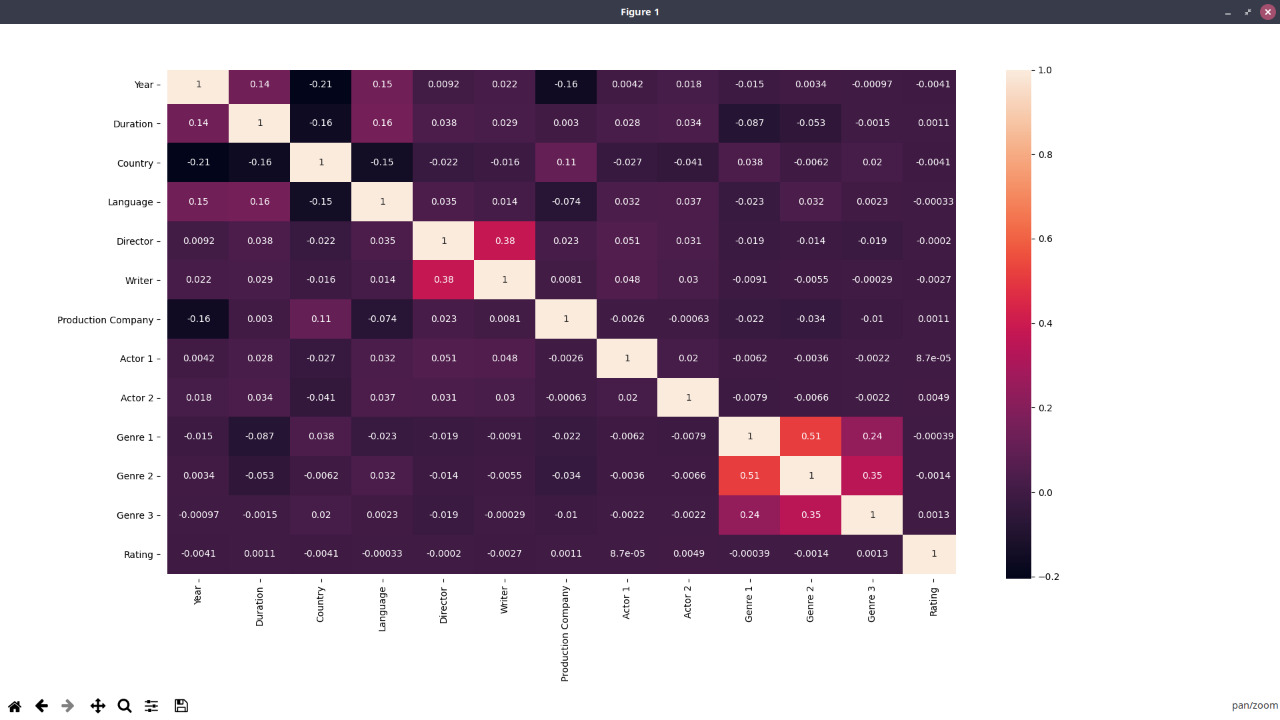


* **Encoding (Converting the Strings into Integer type):**

|  |
| --- |
| dataset['Country']=le.fit\_transform(dataset['Country'])  dataset['Language']=le.fit\_transform(dataset['Language'])  dataset['Director']=le.fit\_transform(dataset['Director'])  dataset['Writer']=le.fit\_transform(dataset['Writer'])  dataset['Production Company']=le.fit\_transform(dataset['Production Company'])  dataset['Actor 1']=le.fit\_transform(dataset['Actor 1'])  dataset['Actor 2']=le.fit\_transform(dataset['Actor 2'])  dataset['Genre 1']=le.fit\_transform(dataset['Genre 1'])  dataset['Genre 2']=le.fit\_transform(dataset['Genre 2'])  dataset['Genre 3']=le.fit\_transform(dataset['Genre 3'])  dataset['Watched']=dataset['Watched'].map({'Yes':1,'No':0}) |

**STARTING DATA CLASSIFICATION (PREDICTION)**

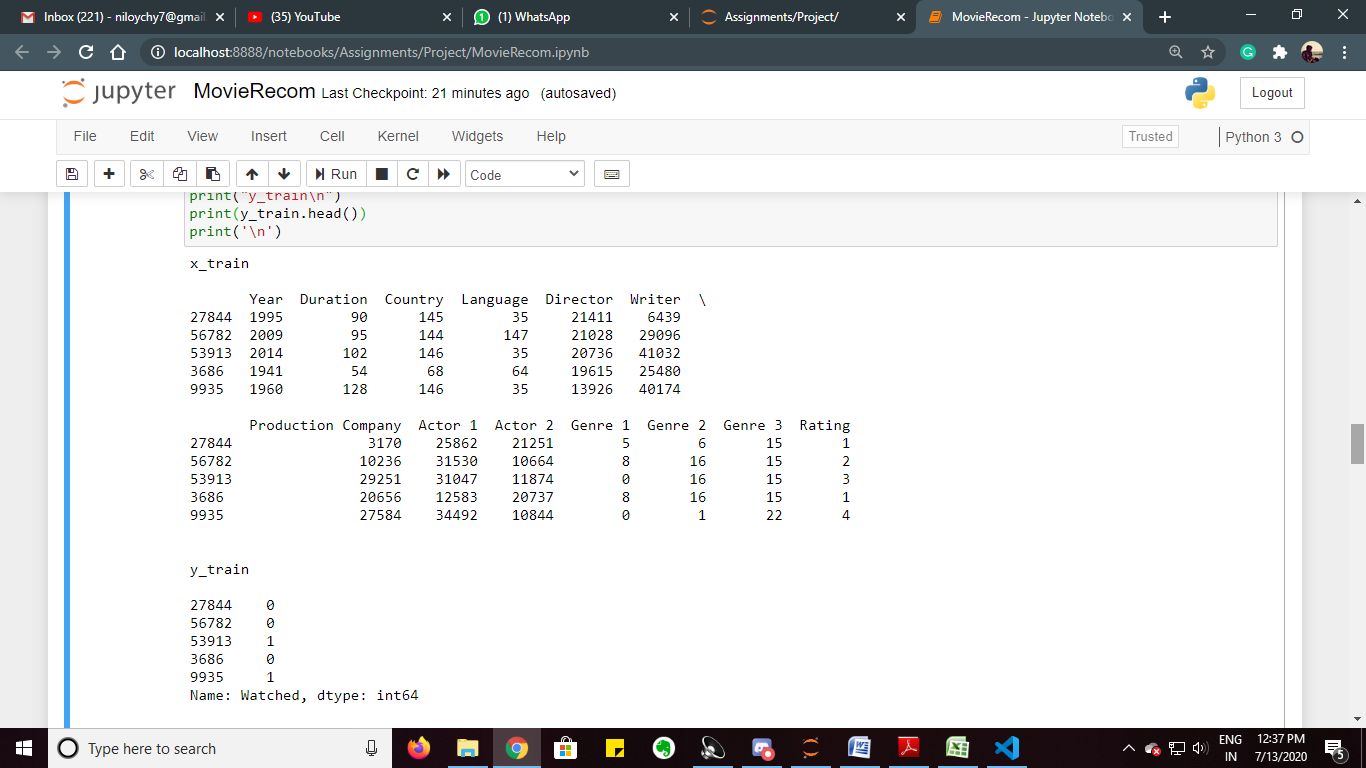
|  |
| --- |
| x=dataset.iloc[:,[1,2,3,4,5,6,7,8,9,10,11,12,13]]  y=dataset.iloc[:,14]  sns.heatmap(x.corr(), annot=True)  plt.show() |



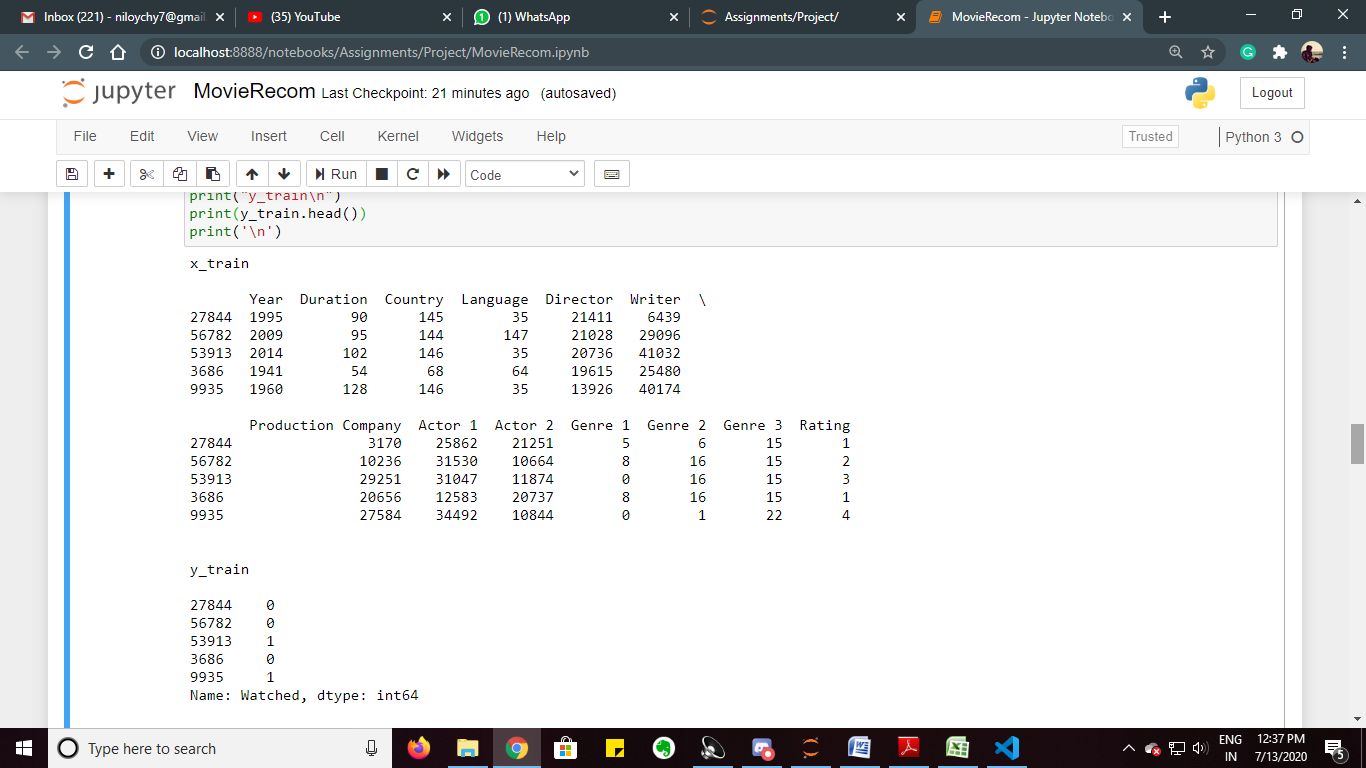
**PREDICTION USING LOGISTIC REGRESSION**

* **Randomly splits the dataset such that test size = 10% (8127 entries) and train size = 90% (73147 entries)**

|  |
| --- |
| x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.1, random\_state = 2029)  print("x\_train\n")  print(x\_train.head()) |

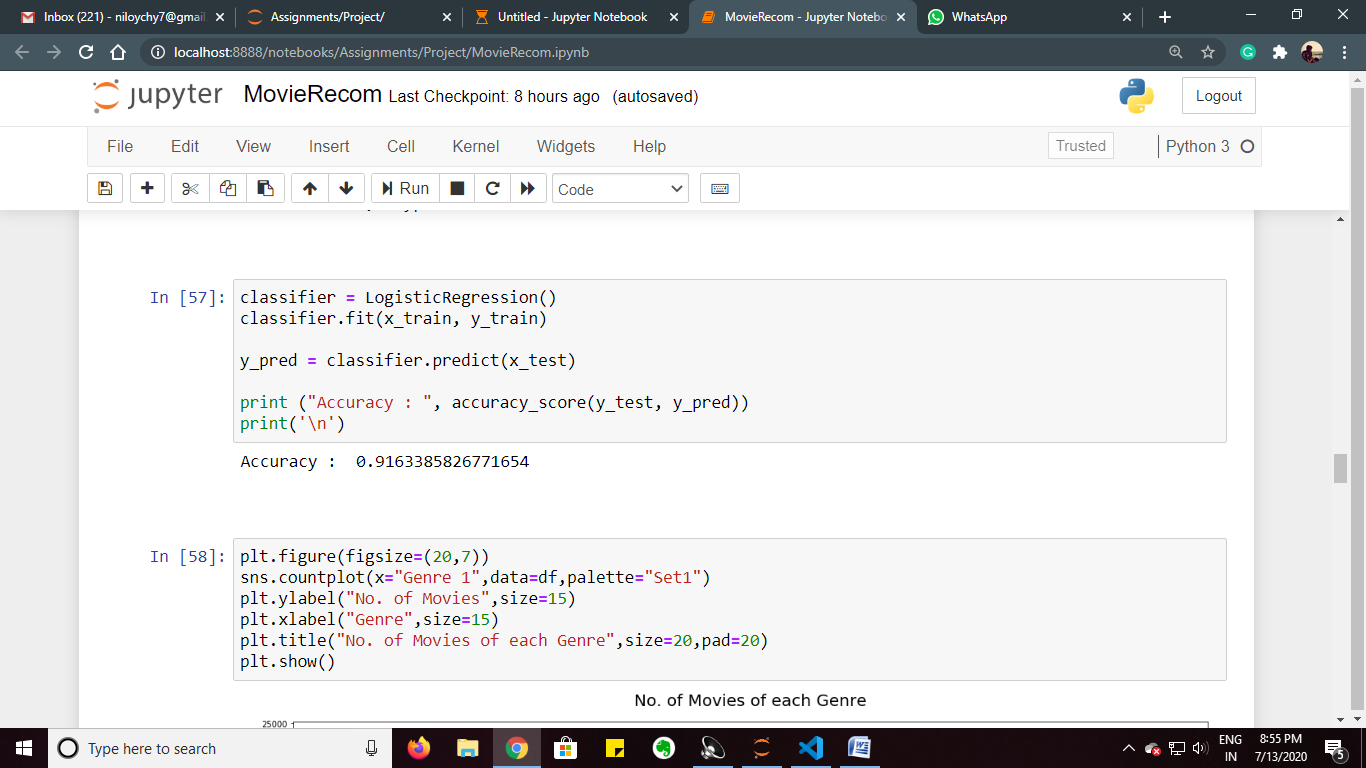


|  |
| --- |
| print("y\_train\n")  print(y\_train.head()) |



* **Using Logistic Regression:**

|  |
| --- |
| classifier = LogisticRegression()  classifier.fit(x\_train, y\_train)  y\_pred = classifier.predict(x\_test)  print ("Accuracy : ", accuracy\_score(y\_test, y\_pred)) |



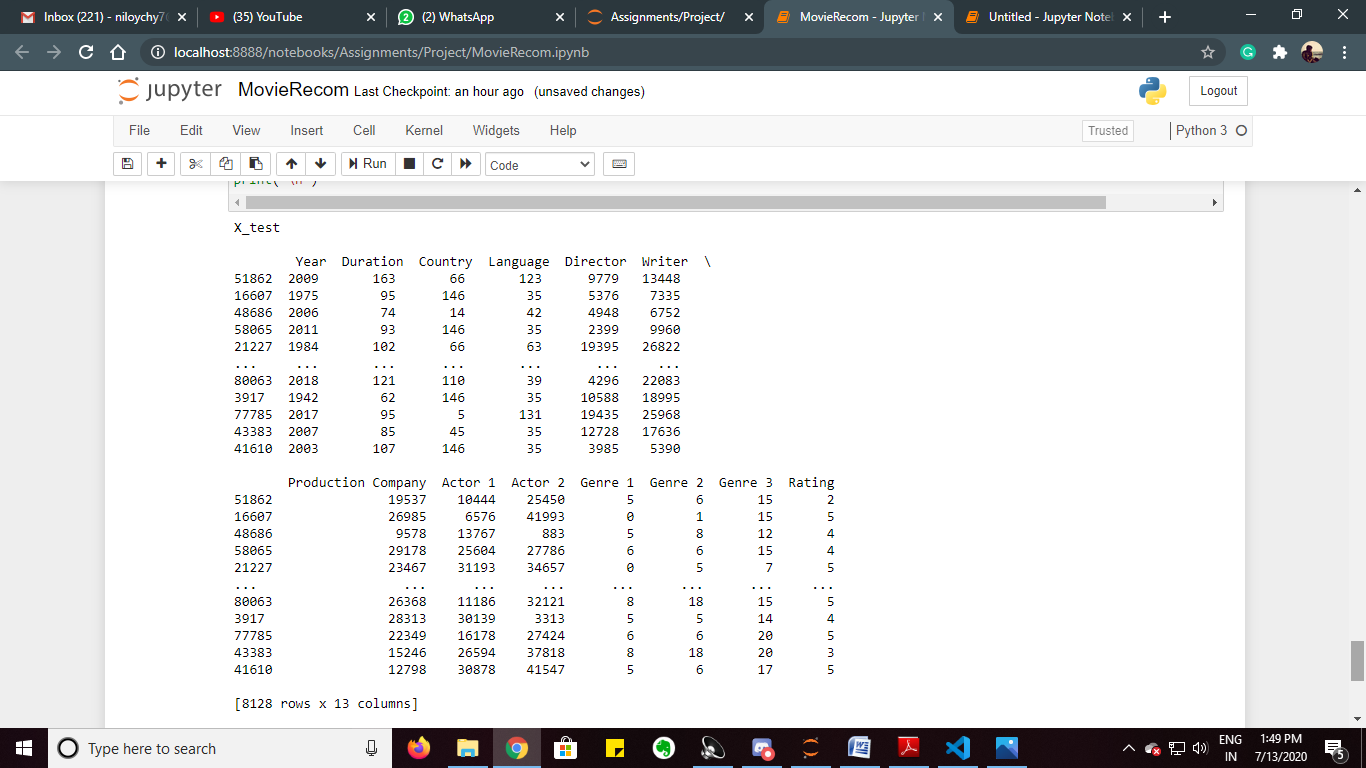
* **For maximum accuracy, Random state can be found out by:**

|  |
| --- |
| maxim=0  p=0  for i in range(900,1000):      x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.1, random\_state = i)      classifier.fit(x\_train, y\_train)      y\_pred = classifier.predict(x\_test)      acc=accuracy\_score(y\_test, y\_pred)      if maxim<acc:          maxim=acc          p=i          print(acc,i)  print(maxim,i) |

**Thus, by using Logistic Regression, we are getting an accuracy of 91.4%.**

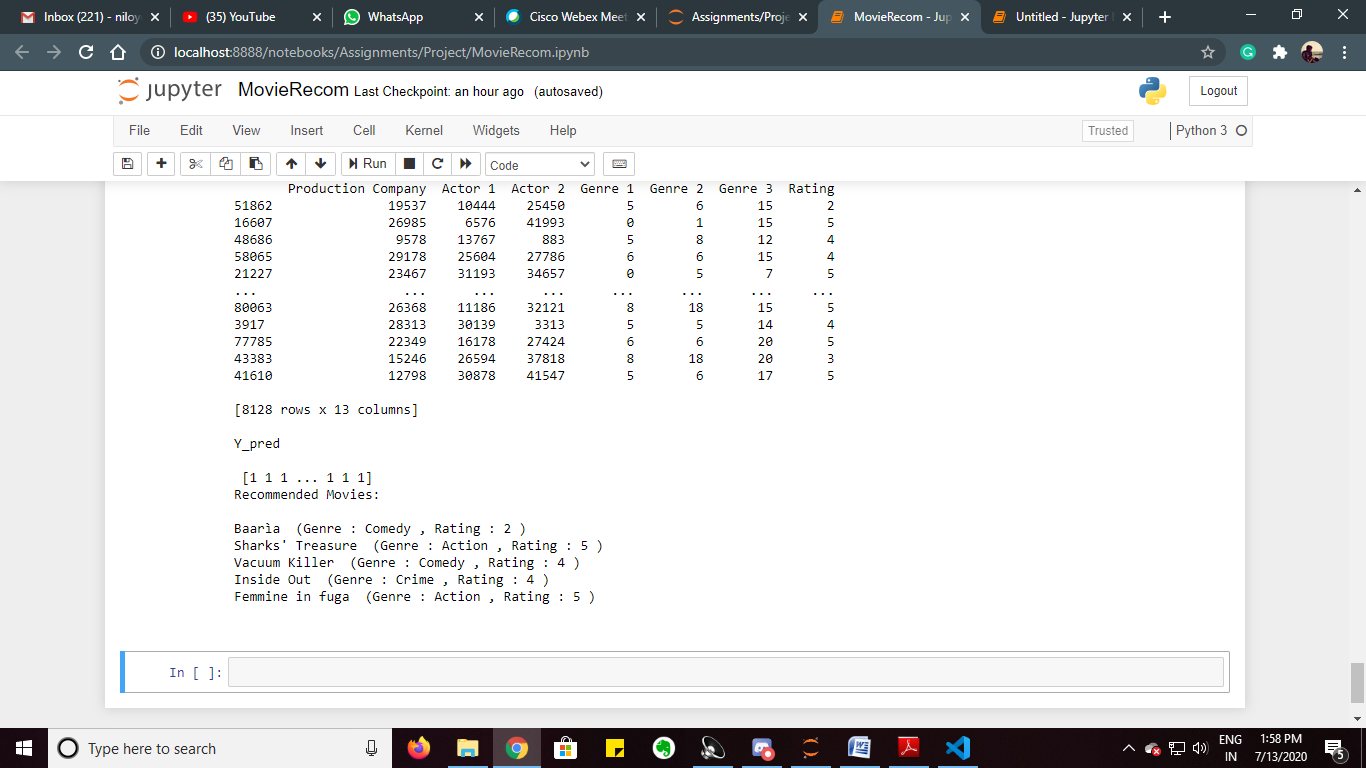
* **x\_test**

|  |
| --- |
| print(x\_test) |



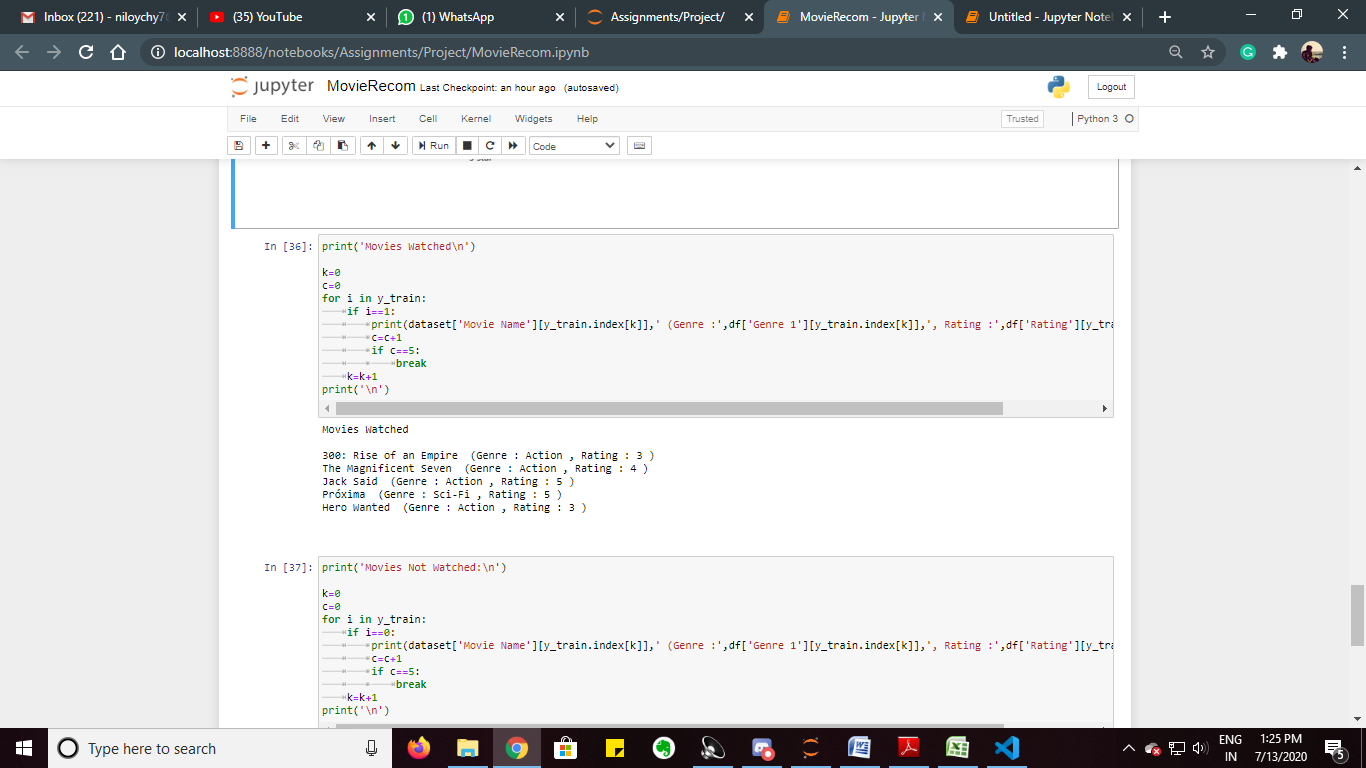
* **y\_pred**

|  |
| --- |
| print(y\_pred) |



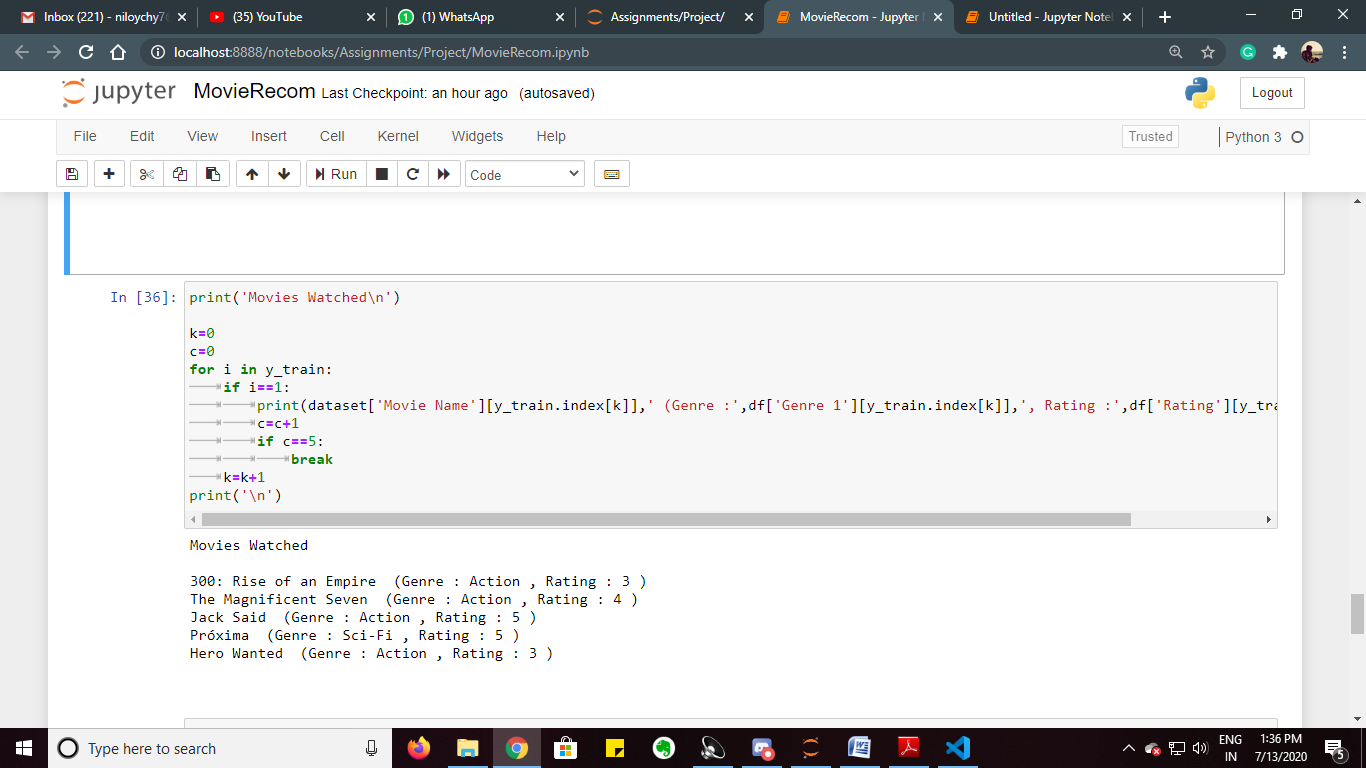
* **Movies from the dataset which are watched:**

|  |
| --- |
| print('Movies Watched\n')  k=0  c=0  for i in y\_train:      if i==1:          print(dataset['Movie Name'][y\_train.index[k]],' (Genre :',df['Genre 1'][y\_train.index[k]],', Rating :',df['Rating'][y\_train.index[k]],')')          c=c+1          if c==5:              break      k=k+1 |



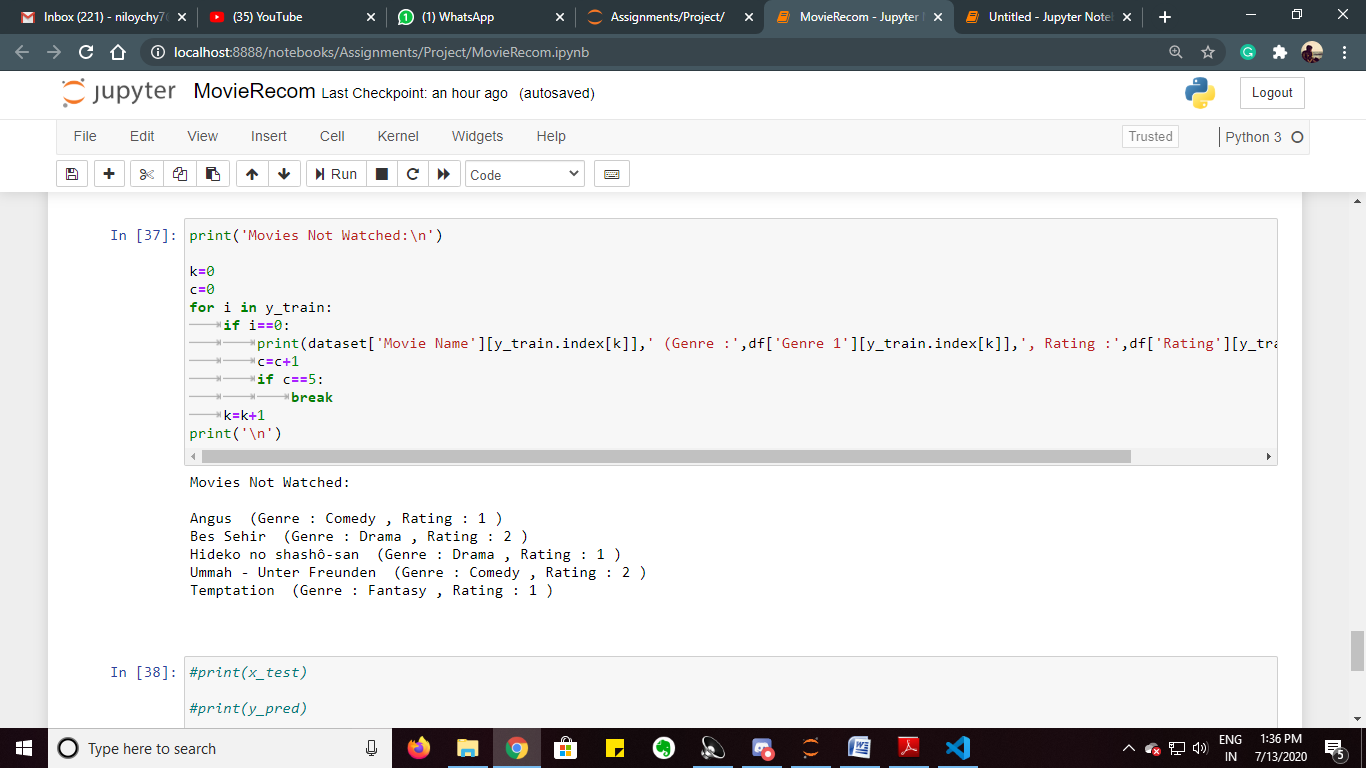
* **Movies from the dataset which are not watched:**

|  |
| --- |
| print('Movies Not Watched:\n')  k=0  c=0  for i in y\_train:      if i==0:          print(dataset['Movie Name'][y\_train.index[k]],' (Genre :',df['Genre 1'][y\_train.index[k]],', Rating :',df['Rating'][y\_train.index[k]],')')          c=c+1          if c==5:              break      k=k+1  print('\n') |



* **Recommend Movies from the dataset:**

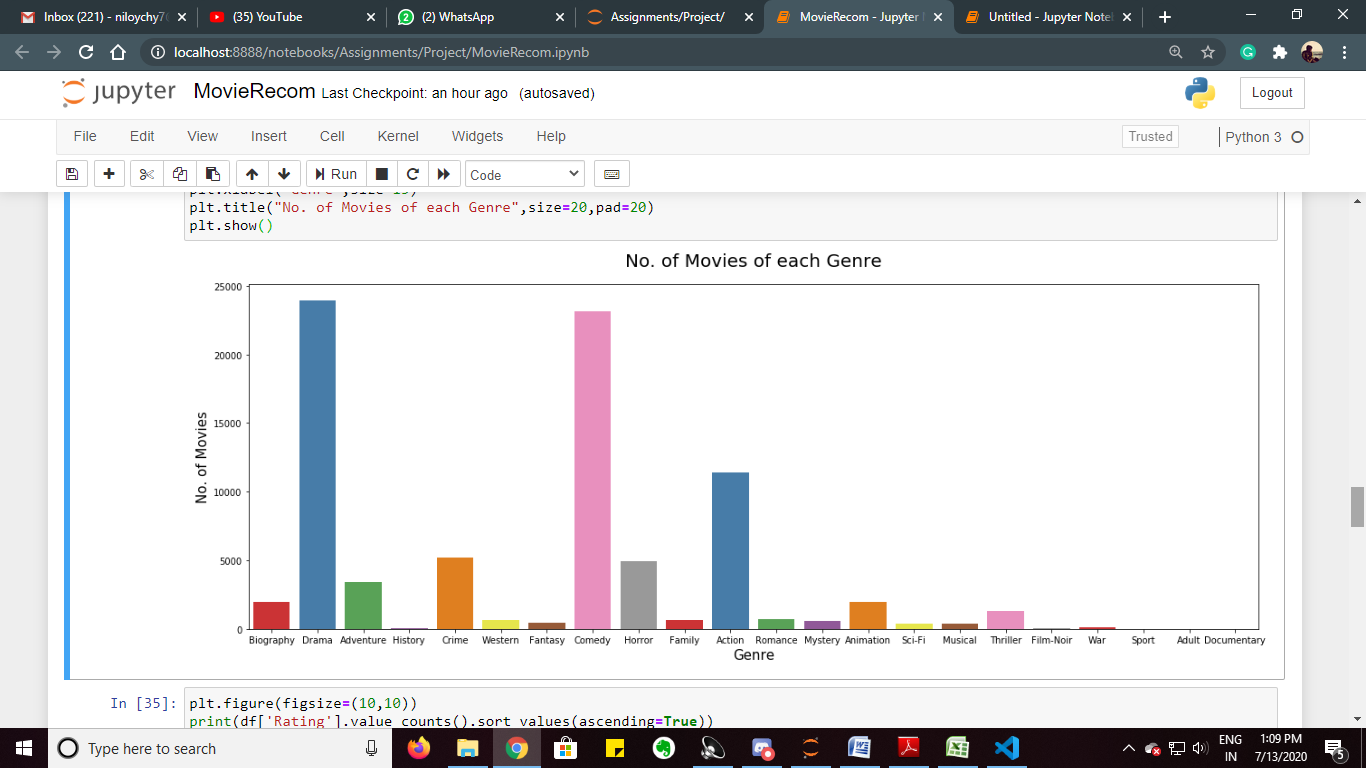
|  |
| --- |
| print('Recommended Movies:\n')  c=0  for i in range(len(y\_pred)):      if y\_pred[i]==1:          print(dataset['Movie Name'][x\_test.index[i]],' (Genre :',df['Genre 1'][x\_test.index[i]],', Rating :',df['Rating'][x\_test.index[i]],')')          c=c+1          if c==5:              break  print('\n') |



**DATA VISUALIZATION**

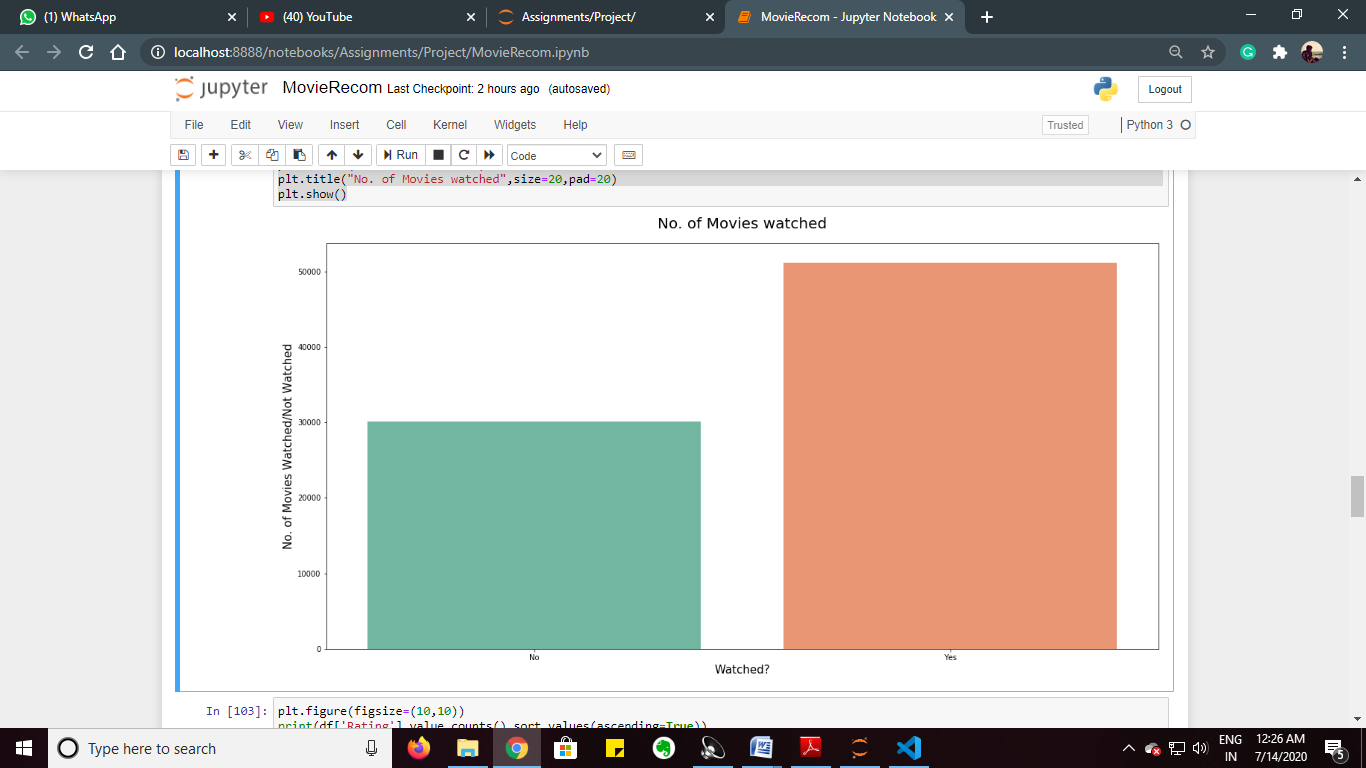
* **No of Movies in each genre:**

|  |
| --- |
| plt.figure(figsize=(20,7))  sns.countplot(x="Genre 1",data=df,palette="Set1")  plt.ylabel("No. of Movies",size=15)  plt.xlabel("Genre",size=15)  plt.title("No. of Movies of each Genre",size=20,pad=20)  plt.show() |



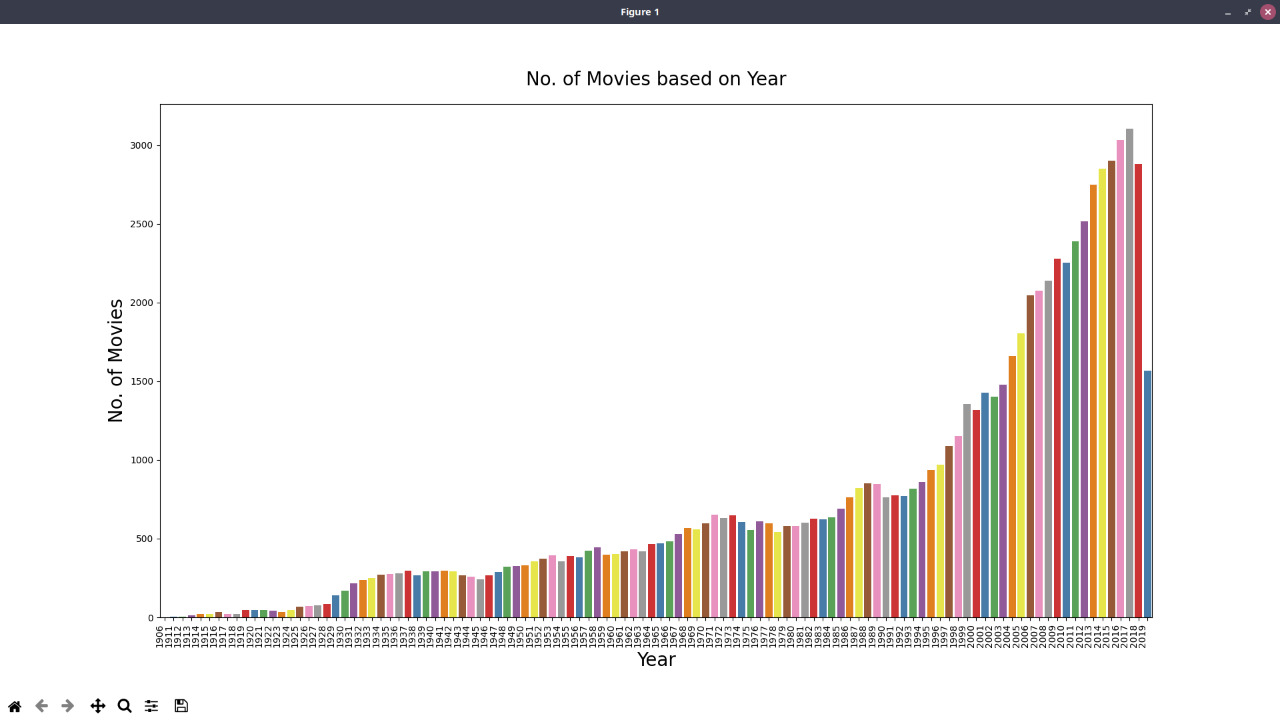
* **No of movies watched:**

|  |
| --- |
| plt.figure(figsize=(20,10))  sns.countplot(x="Watched",data=df,palette="Set2")  plt.ylabel("No. of Movies Watched/Not Watched",size=15)  plt.xlabel("Watched?",size=15)  plt.title("No. of Movies watched",size=20,pad=20)  plt.show() |



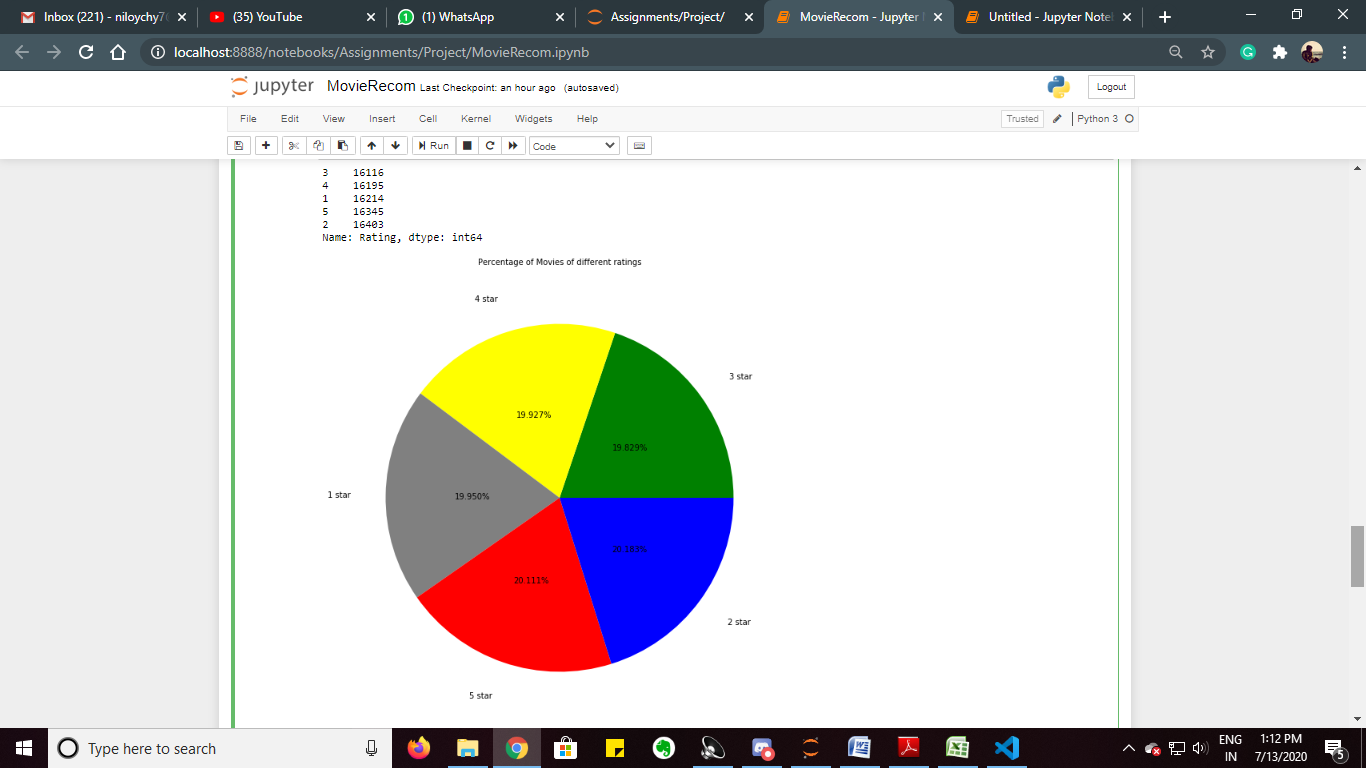
* **No of movies based on Year:**

|  |
| --- |
| plt.figure(figsize=(30,10))  rest = sns.countplot(x="Year",data=df,palette="Set1")  rest.set\_xticklabels(rest.get\_xticklabels(), rotation=90, ha="right")  plt.ylabel("No. of Movies",size=20)  plt.xlabel("Year",size=20)  plt.title("No. of Movies based on Year",size=20,pad=20)  plt.show() |



* **Percentage of movies of different genre**

|  |
| --- |
| plt.figure(figsize=(10,10))  print(df['Rating'].value\_counts().sort\_values(ascending=True))  ratingtype=df['Rating'].value\_counts().sort\_values(ascending=True)  slices=[ratingtype[3],ratingtype[4],ratingtype[1],ratingtype[5],ratingtype[2]]  labels=['3 star','4 star','1 star','5 star','2 star']  colors=['green','yellow','grey','red','blue']  plt.pie(slices,colors=colors,labels=labels,autopct='%1.3f%%',pctdistance=0.5,labeldistance=1.2,shadow=False)  plt.title("Percentage of Movies of different ratings",size=10,pad=20)  plt.show() |



**RESULTS**

ACCURACY OF THE MOVIE RECOMMENDATION SYSTEM IS 91.4%.

**DISCUSSION**

The project uses a dataset of over 81 thousands of movies out of which some of them are watched by the client and the rest are not watched. We randomly break down the dataset into training set (90%) and testing set (10%) such that training set will contain all the features of the movies and consecutively test the prediction onto testing set based on the movies watched earlier and then compare with the original status (of watch/not watched) with the predicted value (of watch/not watched).

**CONCLUSION**

* This project aims at providing the user with movie names based on our watch list of movies.
* It also compares between the actual watched movies and predicted watched movies and thus predicts the recommended movies for the user.
* The accuracy depends on many factors such as rating, genre, language, actors.
* For a user to get recommended movies, he/she must first define what kind of movie he/she earlier watched.
* This project is done with supervised learning algorithm i.e Logistic Regression. The accuracy is further improved by using the particular random state which had the higher accuracy than others.

**REFERENCES**

* <https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset>
* <https://www.coursera.org/learn/machine-learning-with-python>
* Introduction to machine learning with python
* <https://matplotlib.org/>
* <https://numpy.org/>
* <https://scikit-learn.org/stable/>
* <https://pandas.pydata.org/>
* <https://seaborn.pydata.org/>