A

MINI PROJECT REPORT ON

YOUTUBE ANALYSIS USING MACHINE LEARNING

Submitted in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

BY

GAJJALA PRANAYNATH	-	20Q91A05F6
NEERUDI NIKITHA	-	20Q91A05E2
CHOUTAPELLY VARUN	-	20Q91A05C5
CHOWDARY PREETHAM	-	20Q91A05H3

Under the guidance of

Dr.T.SUNIL
DEAN,Dept.of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MALLA REDDY COLLEGE OF ENGINEERING
(Approved by AICTE-Permanently Affiliated to JNTU-Hyderabad)
Accredited by NBA & NAAC, Recognized section 2(f) & 12(B) of UGC New Delhi ISO 9001:2015 certified Institution
Maisammaguda, Dhulapally (Post via Kompally), Secunderabad- 500100

2022 - 2023

MALLA REDDY COLLEGE OF ENGINEERING

(Approved by AICTE-Permanently Affiliated to JNTU-Hyderabad)
Accredited by NBA & NAAC, Recognized section 2(f) & 12(B) of UGC New Delhi ISO 9001:2015 certified Institution
Maisammaguda, Dhulapally (Post via Kompally), Secunderabad- 500100

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the mini project report on "Youtube Analysis Using Machine Learning". is successfully done by the following students of Department of Computer Science and Engineering of our college in partial fulfillment of the requirement for the award of B.Tech Degree in the year of 2023-2024. The results embodied in this report have not been submitted to any other university for the award of any Diploma or Degree.

GAJJALA PRANAYNATH	-	20Q91A05F6
NEERUDI NIKITHA	-	20Q91A05E2
CHOUTAPELLY VARUN	-	20Q91A05C5
CHOWDARY PREETHAM	-	20Q91A05H3

Submitted for the viva voice examination held on : ____________________

INTERNAL GUIDE
Dr.T.SUNIL
DEAN

HOD
Dr.G.RADHA DEVI
Asst.Professor

Internal Examiner

External Examiner

DECLARATION

We, the final year students are hereby declaring that the minor project report titled "Youtube Analysis Using Machine Learning" has done by us under guidance of Mr.T.SUNIL Assistant Professor, Department of CSE is submitted in the partial fulfilment of the requirements for the award of the Degree of **Bachelor of Technology in COMPUTER SCIENCE & ENGINEERING.**

The results embedded in this project report have not been submitted to any other University or Institute for the award of any degree or Diploma.

Signature of the Candidate

GAJJALA PRANAYNATH	**-**	**20Q91A05F6**
NEERUDI NIKITHA	**-**	**20Q91A05E2**
CHOUTAPELLY VARUN	**-**	**20Q91A05C5**
CHOWDARY PREETHAM	**-**	**20Q91A05H3**

DATE:

PLACE: MAISSAMAGUDA

ACKNOWLEDGEMENT

First and foremost, we would like to express our immense gratitude towards our institution Malla Reddy College of Engineering, which helped us to attain profound technical skills in the field of Computer Science & Engineering, there by fulfilling our most cherished goal.

We are pleased to thank Sri Ch. Malla Reddy, our Founder, Chairman **MRGI**, Sri Ch. Mahender Reddy, Secretary, **MRGI** for providing this opportunity and support throughout the course.

It gives us immense pleasure to acknowledge the perennial inspiration of Dr. M. Ashok our beloved principal for his kind co-operation and encouragement in bringing out this task. We would like to thank Dr. T. V. Reddy our vice principal, Dr. G. Radha Devi HOD, CSE Department for their inspiration adroit guidance and constructive criticism for successful completion of our degree.

We would like to thank DR.T.SUNIL Dean our internal guide, for his valuable suggestions and guidance during the exhibition and completion of this project.

Finally, we avail this opportunity to express our deep gratitude to all staff who have contribute their valuable assistance and support making our project success.

GAJJALA PRANAYNATH	**-**	**20Q91A05F6**
NEERUDI NIKITHA	**-**	**20Q91A05E2**
CHOUTAPELLY VARUN	**-**	**20Q91A05C5**
CHOWDARY PREETHAM	**-**	**20Q91A05H3**

ABSTRACT

Social media platforms play a vital role in business, entertainment, marketing, education, media, and communication. YouTube has become the most used platform for sharing videos in society due to its unique behavior. YouTube allows any person to create an account under any category of choosing to upload videos to be viewed by many millions of other people. This has become a trend among the entertainment industry hence it can easily reach to the users and gain popularity for the video materials hosted online.

Many YouTube channel keepers are taking different actions to make the video popular. This research study aims at evaluating the comments provided by the users and identify their requirements and provide recommendations for the You Tubers to make their video popularized. The study has used sentiment analysis and feature extraction methods to derive the set of features required to concern in the development of YouTube videos. Hence the analysis discovers to most important trending videos related to user video types and shows up what are the most trending videos which the user will want to see. The study has used machine learning methods to analyze the trending features and identification of key recommendations for the users. This study has limited to the gaming videos posted in the YouTube channels

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SCREENSHOTS

LIST OF TABLES

LIST OF ABBREVIATIONS

S.NO	SHORT FORM	FULL FORM
1	API	Application Programming Interface
2	CNN	Convolutional Neural Network
3	RNN	Recurrent Neural Network
4	SVM	Support Vector Machine:

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

YouTube has become the second biggest and commonly used search engine and a place where the contributors can earn money by uploading videos. YouTube has gained more than one billion users and influencers as a social media platform within a short period . The users and influencers together have produced billions of perspectives regularly uploading 300 hours of videos constantly. People view around 5 billion videos per day on YouTube. YouTube has created its platform making its viewers hooked up on its variation of categories and income methodologies. The versatility and appealing nature of YouTube have drawn the attention of many people who have different ideas and values. Due to the continuous engagement of different user groups, YouTube has expanded as entertainment, educational, business, social media, and marking platform. As a result of the immense expansion of YouTube, many people around the world have started creating their own YouTube channels. People create YouTube channels, just for fun, educational purposes, gaming, lifestyle, art, technology, vlogging, fashion, exploration and many more. The YouTube channel types mention starts with zero views and subscribers. To analyze or predict content popularity, statistics about viewers, subscribe, watch time and shares must be retrieved. YouTube contains different channels and different categories to provide maximum user experience to the users. One of the key factors to be considered making and pollarding a YouTube video is to make it a trending video. Another important factor is the number of followers. Followers of a channel can bring the YouTube channel to the top level, and every YouTuber prefers to become the top ranked among the trending list. The problem among many Youtubers are lack of awareness on making a video trending. The youtubers should follow suitable methods and practices to increase the views, followers, and the subscriber amount. It is a difficult task for the youtubers to identify a proper mechanism to predict the trending behavior of the channel based on the view’s behavior and comments provided by viewers. As the views of subscriptions and comments are very much useful in popularizing the video channel, this channel. This research aims to devising a method to identify characteristics of trending features of YouTube video channels. The study will predict the characteristic of each video, popularity focusing on the total number of subscribers and the number of views, and the impact of different parameters on the video popularity based on the sentiment classification and analysis of views comments.

An analysis of YouTube data using machine learning involves leveraging computational methods to explore and interpret various aspects of YouTube videos, channels, user interactions, and content trends. This type of project aims to derive insights, predictions, or recommendations from the vast repository of data available on the YouTube platform.

1.2 OBJECTIVES

The primary goal of this project is to apply machine learning techniques to understand, predict, or classify different aspects of YouTube content and user engagement. This might include:

Predicting Video Popularity: Using features like views, likes, comments, and metadata to predict the popularity or virality of videos.

Content Categorization: Classifying videos into different categories or genres based on their content, titles, descriptions, or tags.

User Engagement Analysis: Understanding user behavior through comments sentiment analysis, likes/dislikes patterns, or subscription trends.

Recommendation Systems: Building recommendation algorithms to suggest videos to users based on their preferences, viewing history, or similar content.

1.3 METHODOLOGY:

The methodology consists of three main stages; collect data from YouTube and extract features, sentiment pattern extractor, and sentiment analyzer. In the first stage, the sentiment of YouTube gaming channel comments and feature reviews. In the second stage, sentiment pattern extractor builds all user morphological sentence patterns for sentiment analysis. In the third stage, the sentiment analyzer matches the patterns and sentimental lexicon with the collected data and generate a pattern for the YouTube Video Channels.

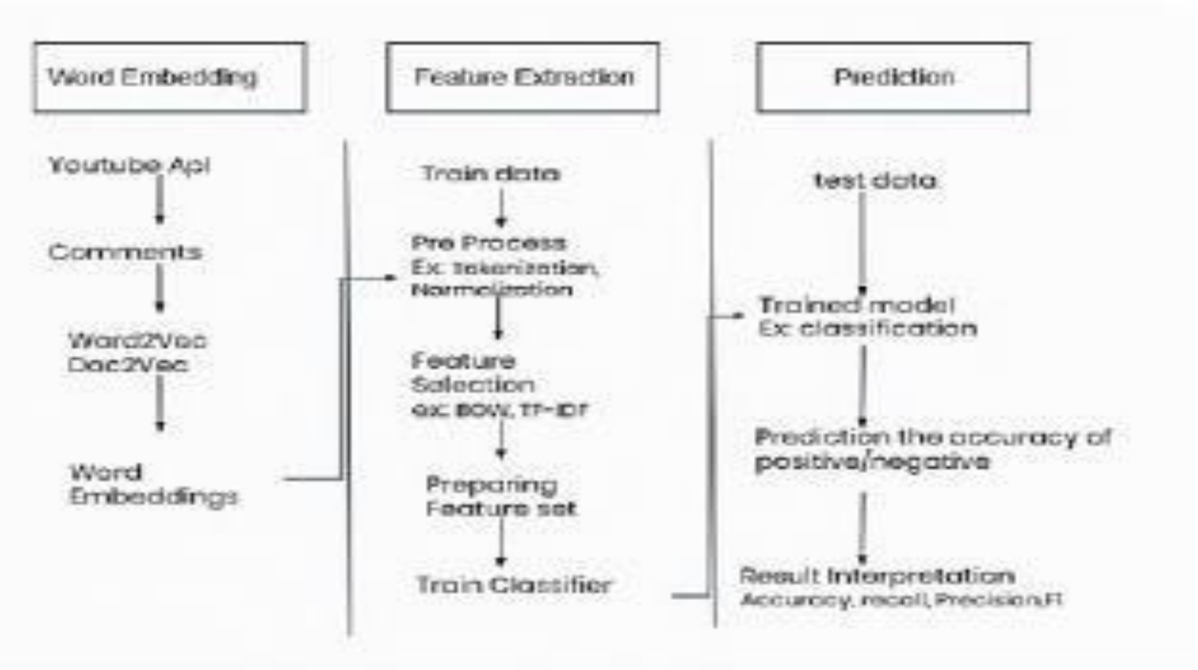


Fig.1: High level architecture of the system

Fig.1 shows the high-level architecture of the research. Fig.1: High level architecture of the system A. Dataset The first is the extraction of comments using the exposed YouTube API v3, and the second being the preprocessing of such comments to prepare for a more effective sentiment calculation. This study collected more than 1000 of the trending YouTube game videos data based on the selected countries from YouTube API V3. Then filter the game videos data from given game category ID and filter those video meta data into CSV format. It was analyzed the user’s opinions use from YouTube videos comments. B. Feature Extraction o f YouTube Channels The data extraction process is done through the programmed written through python language to acquire YouTube videos with a unique ID for each of them with static and dynamic data relevant to certain videos. The feature extraction and the sentiment calculation can be represented in the following Fig 2. Fig. 2: Process of Feature Extraction The feature extraction process is used to derive features from the preprocessed data that can be fed into the machine learning models as the models expect numerical features. Feature extraction can be done in two steps. In the first step, comment specific features are extracted. Furthermore, emoticons/emoji are the relevant comment specific features. Emoticons Can be positive or negative. They are given different weights. Positive emoticons are given a weight of ’1’ and negative emotions are given a weight of’- 1’. There may be positive and negative. Therefore, the count of positive comments and negative comments are added as two separate features in the feature vector. After extracting YouTube comments in specific features, consider the simple text. Feature extraction can be performed by four main methods. Those methods are Bigram model, unigram model, TF-IDF and Word Embedding. Bigram Model: An n-gram is a contiguous sequence of n items from a given sequence of text. Given a sentence, we can construct a list of n-grams from s by finding pairs of words that occur next to each other.(ngram is character based not word-based, and the class does not implement a language model, merely searching for members by string similarity.) Unigram Model: A statistical language model is a probability distribution over sequences of words. TF-IDF: TF-IDF refers the frequency-inverse document frequency. This method is to extract important words from the rest. According to counting terms that have a higher frequency obtains a higher weight, but those words might be frequent in other documents too. Word Embedding: Next process is word embedding and it generates distributed representations. It is performed by embedding word vectors into a continuous vector space according to contextual and semantic similarity. According to the word2vec, it is the technique of embedding the similar sentences of the words using neural network and this method mostly used for increasing the TF- term frequency token

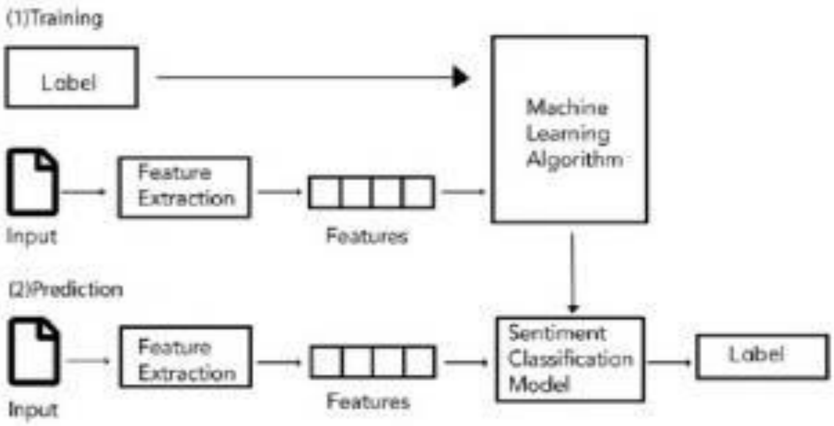


Fig. 2: Process of Feature Extraction

The feature extraction process is used to derive features from the preprocessed data that can be fed into the machine learning models as the models expect numerical features. Feature extraction can be done in two steps. In the first step, comment specific features are extracted. Furthermore, emoticons/emoji are the relevant comment specific features. Emoticons Can be positive or negative. They are given different weights. Positive emoticons are given a weight of ’1’ and negative emotions are given a weight of’- 1’. There may be positive and negative. Therefore, the count of positive comments and negative comments are added as two separate features in the feature vector. After extracting YouTube comments in specific features, consider the simple text. Feature extraction can be performed by four main methods. Those methods are Bigram model, unigram model, TF-IDF and Word Embedding. Bigram Model: An n-gram is a contiguous sequence of n items from a given sequence of text. Given a sentence, we can construct a list of n-grams from s by finding pairs of words that occur next to each other.(ngram is character based not word-based, and the class does not implement a language model, merely searching for members by string similarity.) Unigram Model: A statistical language model is a probability distribution over sequences of words. TF-IDF: TF-IDF refers the frequency-inverse document frequency. This method is to extract important words from the rest. According to counting terms that have a higher frequency obtains a higher weight, but those words might be frequent in other documents too. Word Embedding: Next process is word embedding and it generates distributed representations. It is performed by embedding word vectors into a continuous vector space according to contextual and semantic similarity. According to the word2vec, it is the technique of embedding the similar sentences of the words using neural network and this method mostly used for increasing the TF- term frequency token

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

This paper explains us about concerning many machine learning algorithms are used to predict the performance and backward search is used on features to select out the foremost relevant options. Being a latest kind of job, YouTubers earn cash through the promotional advertisement and bonus from videos. Hence, the recognition of videos is that the prime priority of YouTuber. This project tries to predict the working of the videos that are going to be uploaded to YouTube. Predicting quality of social media contents has attracted wide attention in recently years. The dimensions of gigantic database make machine learning become a sturdy tool to deal with the matter. This project explores the way to use machine learning algorithm to predict the video performance for YouTuber. Once mistreatment several algorithms, model enhancements, and backward search on features. Thus, in this paper we have a tendency to formalize the matter of predicting trends and hits in user generated videos. Also, we have a tendency to describe our research and analysis methodology on approaching this problem. To the simplest of information, our work is novel in that specializes in the problem of predicting popularity trends complementary to hits. Moreover, we have a tendency to intend on evaluating effectiveness of our results not solely based on common applied statistical error metrics, however conjointly on the attainable online advertising revenues our predictions can generate. Once describing our proposal, we have a tendency to here summarize our latest findings regarding to uncovering common popularity trends, measuring associations between UGC features and recognition trends, and assessing the effectiveness of models for predicting quality trends.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

YouTube acts as an organizes field where YouTube gamers play the role of actors . In present YouTube has become a social media leader in sharing and provide the content creators with videos . They successfully gain the monetary profits from the views and their contents. YouTube encourages creators to publish their videos and allow others to comment on the videos. The longer videos have the greatest number of comments most of the time. The higher number of comments, shares, and views make the video go high the ranking system will keep the channel trending . YouTube has an average association between a channel's subscriber size and rankings

3.2 EXISTING SYSTEM DISADVANTAGES

1.LESS ACCURACY

2.LOW EFFICENCY

3.3 PROPOSED SYSTEM

In this experiment, our main objective is to predict the potential total view count of a YouTube video as accurate as possible based on several influential factor. To do so, we have decided to apply one of the predictive modeling techniques, which is the regression technique in our experiment. We will be using regression technique to model the mathematical correlation between our independent variables, which are the attributes in the dataset we have acquired, and the dependent variable, in this case it is the amount of viewership of a YouTube video. After figuring out the pattern and the relationship between the said variables, we are then able to predict the future value of the dependent variable. One of the key benefits that regression analysis offers is that it indicates the strength of impact of multiple independent variables on a dependent variable. This will allow us to compare the outcome when a variable has its value changed. For example, the result of this experiment will show how the channel subscriber count will affect the total view count of a YouTube video.

K Neighbors Classifier K-nearest neighbors could be a straightforward algorithmic program that stores all accessible cases and classifies new cases supported a similarity measure knearest neighbors' algorithmic program (k-NN) could be a nonparametric technique used for classification and regression. In each case, the input consists of the k-nearest coaching examples within the feature area. The output depends on whether k-NN is employed for classification or regression: In k-NN classification, the output could be a category membership. An object is assessed by a plurality vote of its neighbors, with the item being appointed to the category commonest among its k nearest neighbors (k could be a positive whole number, usually small). If k = 1, then the object is solely appointed to the category of that single nearest neighbor. In k-NN regression, the output is that the property price for the item. This price is that the average of the values of k nearest neighbors. KNN could be a variety of instancebased learning, or lazy learning, wherever operate is simply approximated regionally and every one computation is deferred till classification. The k-NN algorithmic program is among the only of all machine learning algorithms. Both for classification and regression, a helpful technique may be accustomed assign weight to the contributions of the neighbors, so that the nearer neighbors contribute additional to the common than the additional distant ones. As an example, a typical weight theme consists in giving every neighbor a weight of 1/d, wherever d is that the distance to the neighbor.

Decision Tree Regression In this algorithm at a specific node, a split of knowledge happens. Thus, the most effective attribute to separate is to be known. Once the split for every price, a child node is formed. For every child node if the set is pure then it might stop otherwise a algorithmic split happens. This algorithm is prime down and goes on in dividend conquer manner. There are three varieties of nodes in an exceedingly call tree specifically root node, branch node and leaf node. Leaf node represents a category. Partitioning of knowledge is stopped once the subsequent conditions are satisfied.

3.4 PROPOSED SYSTEM ADVANTAGES

1.HIGH ACCURACY

2.HIGH EFFICENCY

3.5 HARDWARE & SOFTWARE REQUIREMENTS

3.5.1 HARD REQUIRMENTS :

- System : i3 or above.
- Ram : 4 GB.
- Hard Disk : 40 GB

3.5.2 SOFTWARE REQUIRMENTS :

- Operating system : Windows8 or Above.
- Coding Language : python

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

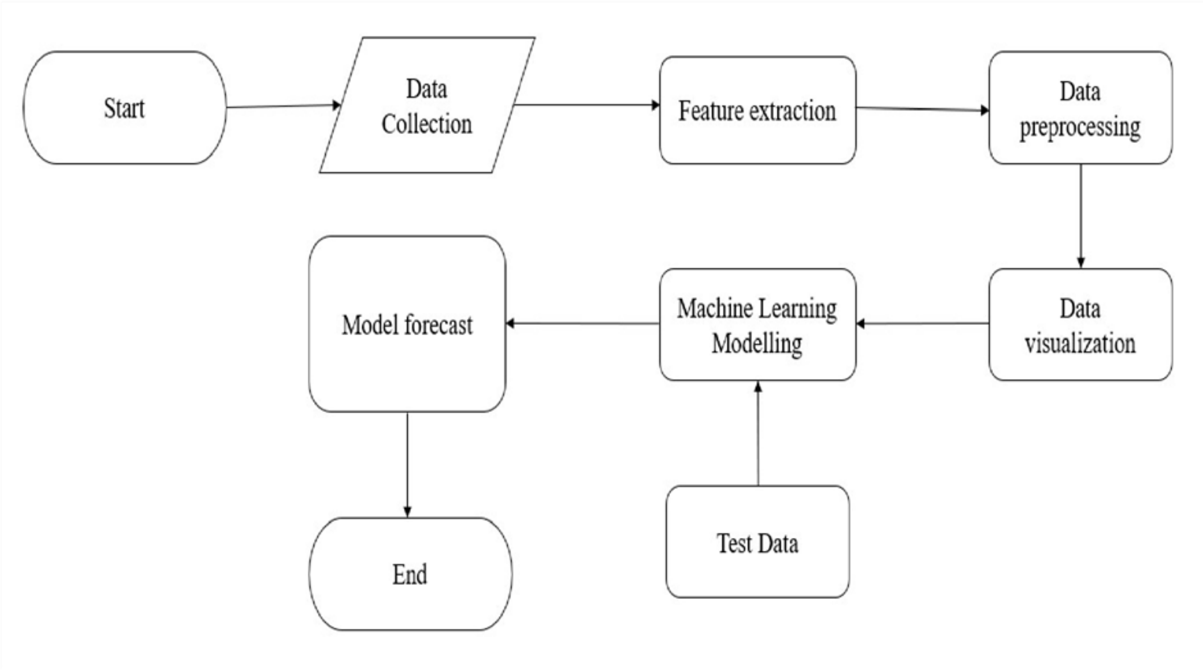


Fig 4.1.1: System Architecture.

4.2 MODULES

- Data Collection
- Data Pre-processing
- Feature Extraction
- Evaluation Model

4.2.1 Data Collection

Data collection is the process of gathering information from various sources, which is later used to develop machine learning models. The data should be stored in a way that makes sense for the problem at hand. In this step, the dataset is converted into an understandable format that can be fed into machine learning models.

The data used in this paper consists of a set of data with a few features. This step focuses on selecting the subset of all available data that will be used for the project. Machine learning problems typically require a large amount of data, preferably labeled data, which means data for which the target answer is already known. Our input data will be collected in a dataset format, which consists of past records relevant to the problem statement.

4.2.2 Data Pre-processing

To organize the selected data, three common data pre-processing steps need to be performed: formatting, cleaning, and sampling.

Formatting: The selected data may not be in a suitable format for analysis. It could be stored in a relational database while you prefer a flat file format, or it may be in a proprietary file format that you want to convert to a more accessible format like a relational database or a text file.

Cleaning: Cleaning the data involves dealing with missing or incomplete data. Some instances in the data may lack necessary information to address the problem at hand, and those instances may need to be removed. Additionally, if certain attributes contain sensitive information, they may need to be anonymized or completely removed from the dataset. Sampling: In cases where the selected data is extensive, it may be more practical to work with a smaller representative sample. Having a large amount of data can significantly increase computational and memory requirements, as well as the running times of algorithms. By taking a smaller sample, you can explore and prototype solutions more efficiently before considering the entire dataset.

By performing these pre-processing steps, the selected data can be organized, formatted, cleaned of inconsistencies or missing values, and sampled appropriately for further analysis and modeling.

4.2.3 Feature Extraction

Next, feature extraction is performed as an attribute reduction process. Unlike feature selection, which ranks existing attributes based on their predictive significance, feature extraction involves transforming the attributes themselves. The transformed attributes, known as features, are linear combinations of the original attributes. After feature extraction, our models are trained using a classifier algorithm. In our case, we utilize the classify module from the Natural Language Toolkit library in Python. We employ a labeled dataset that we have gathered, reserving a portion of the labeled data for evaluating the models.

Various machine learning algorithms were employed to classify the pre-processed data, with the chosen classifiers being Random Forest, Decision Tree, SVM. These algorithms are

4.2.4 Evaluation Model

Model evaluation is an integral part of the model development process as it helps identify the best model to represent our data and assess how well the chosen model will perform in the future. Evaluating model performance using the same data used for training is not acceptable in data science as it can lead to overoptimistic and overfitted models. The performance of each classification model is estimated by calculating its average. The results are then visualized, typically in the form of graphs, to represent the classified data. Accuracy is a commonly used metric that measures the percentage of correct predictions for the test data. It is calculated by dividing the number of correct predictions by the total number of predictions.

Once the ML algorithm is chosen and the training data is prepared, it undergoes the algorithmic process, resulting in predicted output. This predicted output is then compared with the test data to evaluate the model's performance.

In summary, model evaluation is essential for selecting the best model and assessing its future performance. It involves using separate test data to avoid overfitting, calculating performance metrics such as accuracy, and visualizing the results through graphs or other visual representations.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 MACHINE LEARNING

Machine Learning is a system that can learn from examples through self-improvement and without being explicitly coded by the programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., an example) to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to make actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The machine receives data as input and uses an algorithm to formulate answers.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendations.

Machine learning is also used for a variety of tasks like fraud detection, predictive maintenance, portfolio optimization, automated task, and so on.

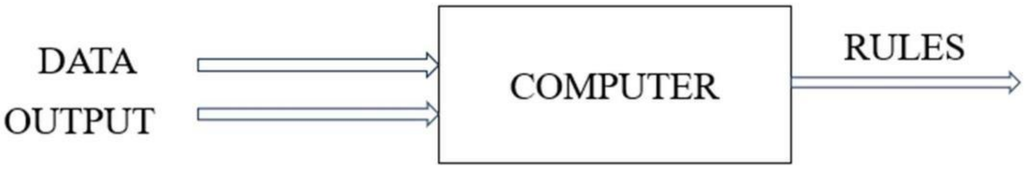


Fig 5.1.1: Machine Learning.

5.1.1 Working of Machine Learning

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feeds a previously unseen example, the machine has difficulties predicting.

The core objective of machine learning is the learning and inference. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the data.

One crucial part of the data scientist is to choose carefully which data to provide to the machine.

The list of attributes used to solve a problem is called a feature vector. You can think of a feature vector as a subset of data that is used to tackle a problem. The machine uses some fancy algorithms to simplify the reality and transform this discovery into a model. Therefore, the learning stage is used to describe the data and summarize it into a model.

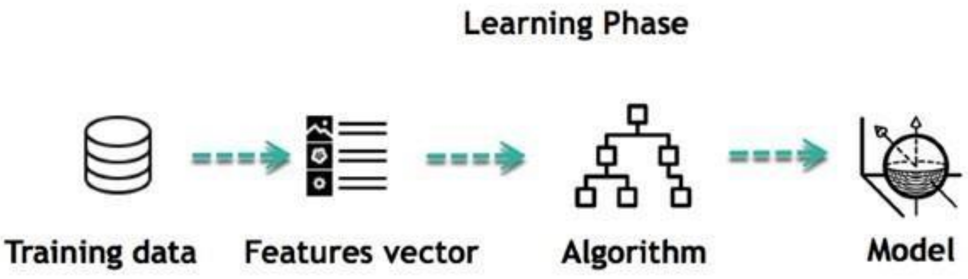


Fig 5.1.2: Learning phase of Machine Learning.

For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant. This is the model inferring.

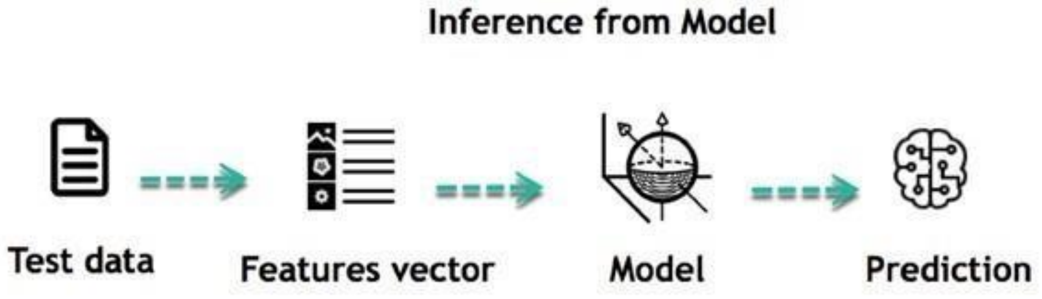


Fig 5.1.3: Machine Learning Inference from Model.

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

The life of Machine Learning programs is straightforward and can be summarized in the following points:

- Define a question
- Collect data
- Visualize data
- Train algorithm
- Test the Algorithm
- Collect feedback
- Refine the algorithm
- Loop 4-7 until the results are satisfying
- Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

5.1.2 Types of Machine Learning Algorithms

Machine learning can be grouped into three broad learning tasks: Supervised, Unsupervised, and Reinforcement Learning.

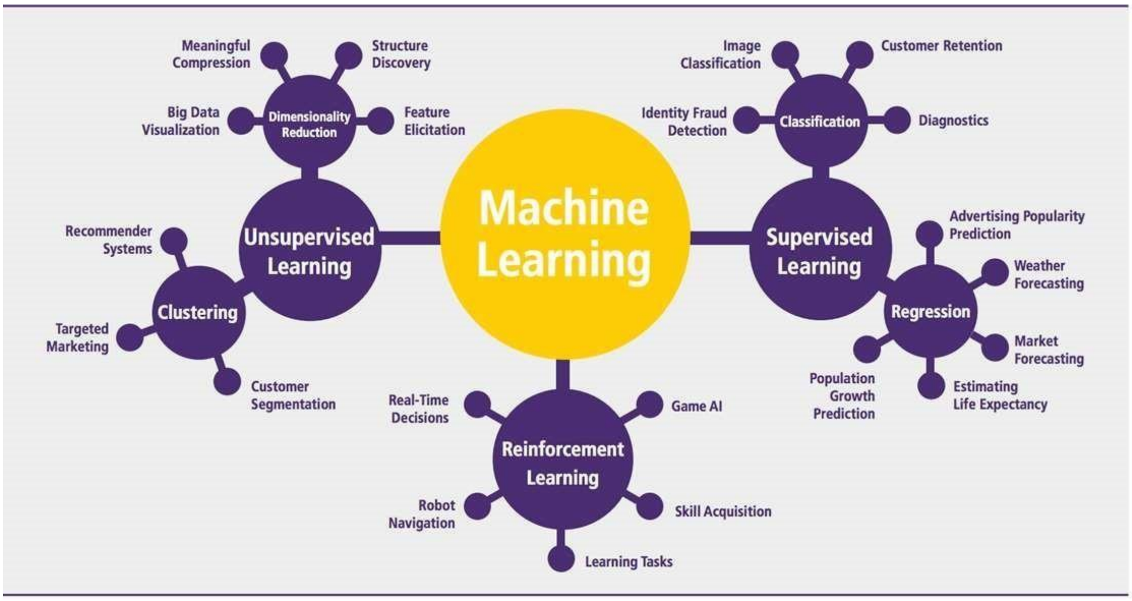


Fig 5.1.4: Types of Machine Learning Algorithms

1. Supervised learning

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans. You can use supervised learning when the output data is known. The algorithm will predict new data.

Supervised Learning includes the following algorithms:

- Linear Regression
- Logistic Regression
- Decision Tree
- Naïve Bayes
- Support Vector Machine
- Random Forest
- AdaBoost
- Gradient Boosting Trees

There are two categories of supervised learning:

- **Classification**

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female. The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

- **Regression**

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

2. Unsupervised learning

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns) You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you.

Unsupervised Learning includes following algorithms:

- K-means clustering
- Gaussian mixture model
- Hierarchical clustering
- Recommender system
- PCA/T-SNE

3. Reinforcement Learning

Reinforcement learning is a subfield of machine learning in which systems are trained by receiving virtual "rewards" or "punishments," essentially learning by trial and error. Google's DeepMind has used reinforcement learning to beat a human champion in the Go games. Reinforcement learning is also used in video games to improve the gaming experience by providing smarter bots.

One of the most famous algorithms is:

- Q-learning

- Deep Q network
- State-Action-Reward-State-Action (SARSA) □ Deep Deterministic Policy Gradient (DDPG)

5.2 PYTHON

Python programming language is used for building the machine learning model.

5.2.1 Introduction

Python is an object-oriented, high level language, interpreted, dynamic and multipurpose programming language. Python is easy to learn yet powerful and versatile scripting language which makes it attractive for Application Development. Python's syntax and dynamic typing with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas.

Python supports multiple programming pattern, including object oriented programming, imperative and functional programming or procedural styles. Python is not intended to work on special area such as web programming. That is why it is known as multipurpose because it can be used with web, enterprise, 3D CAD etc. We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to declare an integer value in a variable. Python makes the development and debugging fast because there is no compilation step included in python development and edit-test-debug cycle is very fast.

5.2.2 Python Features

1) Easy to Use

Python is easy to very easy to use and high-level language. Thus it is a programmer-friendly language.

2) Interpreted Language

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

3) Cross-platform language

Python can run equally on different platforms such as Windows, Linux, Unix, Macintosh etc.

Thus, Python is a portable language.

4) Free and Open Source

Python language is freely available.

5) Object-Oriented language

Python supports object-oriented language. The concept of classes and objects comes into existence.

6) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in your Python code.

7) Large Standard Library

Python has a large and broad library.

8) GUI Programming

Graphical user interfaces can be developed using Python.

9) Integrated

It can be easily integrated with languages like C, C++, JAVA etc.

5.2.3. Python History

Python laid its foundation in the late 1980s.The implementation of Python was started in the December 1989 by Guido Van Rossum at CWI in Netherland. ABC programming language is said to be the predecessor of Python language which was capable of Exception Handling and interfacing with Amoeba Operating System.

5.2.4. Python Version

Python programming language is being updated regularly with new features and support. There are a lot of updates in python versions, started from 1994 to current date. A list of python versions with its released date is given below

Python Version	Released Date
Python 1.0	January 1994
Python 1.5	December 31, 1997
Python 1.6	September 5, 2000
Python 2.0	October 16, 2000
Python 2.1	April 17, 2001
Python 2.2	December 21, 2001
Python 2.3	July 29, 2003
Python 2.4	November 30, 2004
Python 2.5	September 19, 2006
Python 2.6	October 1, 2008
Python 2.7	July 3, 2010
Python 3.0	December 3, 2008
Python 3.1	June 27, 2009
Python 3.2	February 20, 2011
Python 3.3	September 29, 2012

Table 5.2.1: Python Version Table.

5.2.5. Python Applications

Python as a whole can be used in any sphere of development. Let us see what are the major regions where Python proves to be handy.

1) Console Based Application

Python can be used to develop console based applications.

2) Audio or Video based Applications

Python proves handy in multimedia section.

3) 3D CAD Applications

Fandango is a real application which provides full features of CAD.

4) Web Applications

Python can also be used to develop web based application. Some important developments are: PythonWikiEngines, Pocoo, PythonBlogSoftware etc.

5) Enterprise Applications

Python can be used to create applications which can be used within an Enterprise or an Organization.

6) Applications for Images

Using Python several application can be developed for image. Applications developed are: VPython, Gogh, imgSeek etc. There are several such applications which can be developed using Python

5.2.6 Python Execution

1) Interactive Mode:

You can enter “python” in the command prompt and start working with Python by executing Python commands.

2) Script Mode:

Using Script Mode , Python code is written in a separate file using any editor of the Operating System. It is then saved using .py extension. In order to open use the command “ python file_name.py” after setting the path of the file in command prompt.

NOTE: Path in the command prompt should be where you have saved your file. In the above case file should be saved at desktop.

3) Using IDE: (Integrated Development Environment)

Python code can be executed using a Graphical User Interface (GUI).It is done by following the below steps:

Click on Start button -> All Programs -> Python -> IDLE(Python GUI) In IDE both interactive and script mode can be used.

□ Using Interactive mode:

Execute your Python code on the Python prompt and it will display result simultaneously. □ Using Script Mode:

i) Click on Start button -> All Programs -> Python -> IDLE(Python GUI) ii) Python Shell will be opened. Now click on File -> New Window.

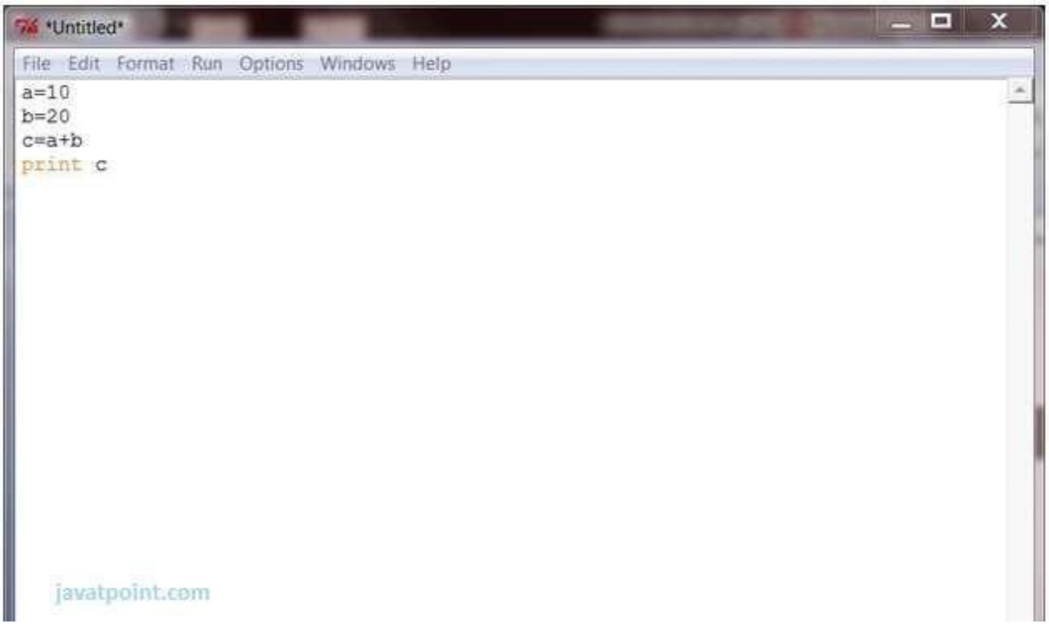
A new Editor will be opened . Write your Python code here.

Click on file -> save as

Run then code by clicking on Run in the Menu bar.

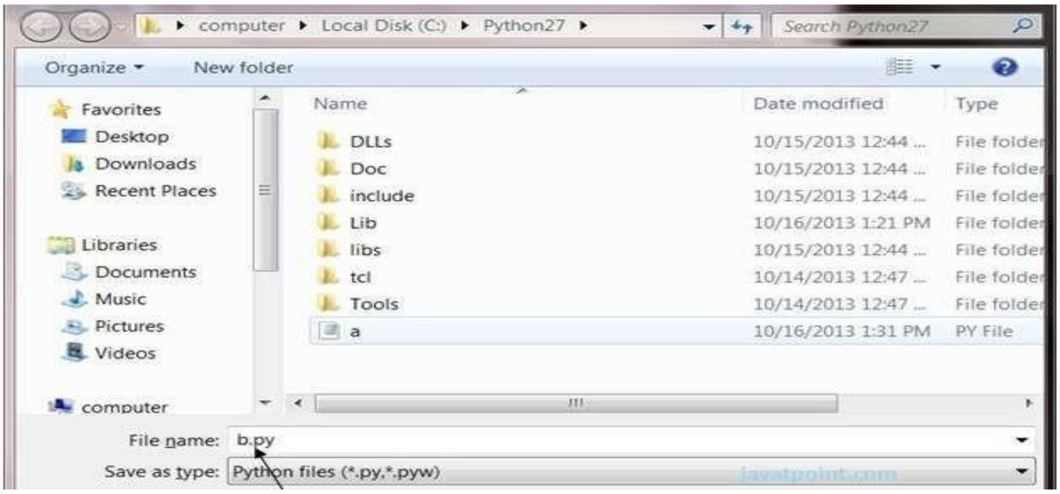
Run -> Run Module

Result will be displayed on a new Python shell



```
a=10
b=20
c=a+b
print c
```

(a)

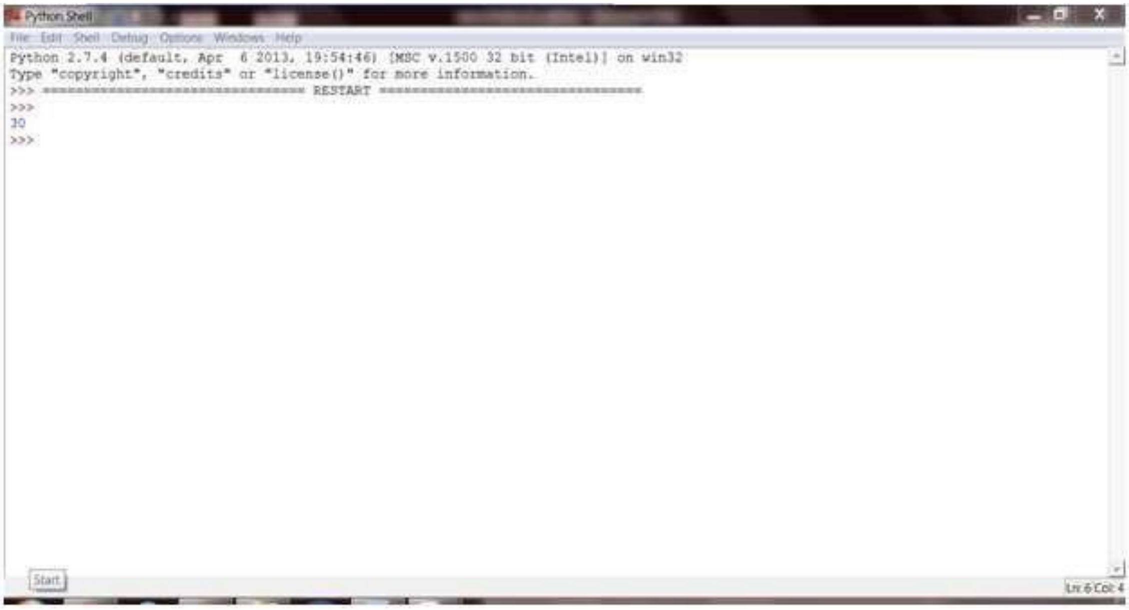


Name	Date modified	Type
DLLs	10/15/2013 12:44 ...	File folder
Doc	10/15/2013 12:44 ...	File folder
include	10/15/2013 12:44 ...	File folder
Lib	10/16/2013 1:21 PM	File folder
libs	10/15/2013 12:44 ...	File folder
tcl	10/14/2013 12:47 ...	File folder
Tools	10/14/2013 12:47 ...	File folder
a	10/16/2013 1:31 PM	PY File

File name: b.py

Save as type: Python files (*.py,*.pyw)

(b)



```
Python 2.7.4 (default, Apr  6 2013, 19:54:46) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
30
>>>
```

(c)

Fig 5.2.1: Python program execution in IDE script mode.

5.2.7 Python Variables

Variable is a name of the memory location where data is stored. Once a variable is stored that means a space is allocated in memory. For assigning values to variable we need not to declare explicitly variable in Python. When we assign any value to the variable that variable is declared automatically. The assignment is done using the equal (=) operator.

5.2.8 Assigning variables

1. Assigning single value to multiple variables:

Eg: x=y=z=50

2.Assigning multiple values to multiple variables:

Eg: x, y, z=10,20,30

The values will be assigned in the order in which variables appears.

5.3 FUNDAMENTALS OF PYTHON

This section contains the basic fundamentals of Python.

5.3.1 Tokens

Tokens can be defined as a punctuator mark, reserved words and each individual word in a statement. Token is the smallest unit inside the given program. Tokens include Keywords, Identifiers, Literals, Operators.

5.3.2 Tuples

Tuple is another form of collection where different type of data can be stored. It is similar to list where data is separated by commas. Only the difference is that list uses square bracket and tuple uses parenthesis. Tuples are enclosed in parenthesis and cannot be changed. Eg: tuple=('rahul',100,60.4,'deepak')

5.3.3 Dictionary

Dictionary is a collection which works on a key-value pair. It works like an associated array where no two keys can be same. Dictionaries are enclosed by curly braces ({}) and values can be retrieved by square bracket([])

Eg: dictionary={'name':'charlie','id':100,'dept':'it'}

5.4 JUPYTER NOTEBOOK

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It provides an interactive computing environment that supports various programming languages, including Python, R, Julia, and more. Here are some key notes on Jupyter Notebook:

1. User-Friendly Interface: Jupyter Notebook has a user-friendly interface that allows you to create and organize your code in cells. You can easily write and execute code, view the output, and add explanatory text using Markdown.

2. Code Execution: Jupyter Notebook provides an interactive environment where you can execute code in individual cells. This allows for iterative development and easy debugging. You can execute cells individually or run the entire notebook.

3. Mix of Code and Text: One of the unique features of Jupyter Notebook is the ability to mix code and text in the same document. You can write explanatory text using Markdown syntax, add images, and even mathematical equations using LaTeX.

4. Data Visualization: Jupyter Notebook supports the integration of data visualization libraries such as Matplotlib, Seaborn, and Plotly. You can generate interactive plots, charts, and graphs directly in the notebook, making it easy to explore and analyze data.

5. Collaboration and Sharing: Jupyter Notebook allows you to share your work with others by exporting notebooks to various formats, including HTML, PDF, and slides. You can also collaborate with team members by sharing the notebook files and working together in real-time using platforms like JupyterHub or JupyterLab.

6. Rich Ecosystem: Jupyter Notebook is part of a rich ecosystem of tools and libraries for scientific computing and data analysis in Python. It integrates well with popular libraries such as NumPy, Pandas, SciPy, and scikit-learn, making it a powerful tool for data scientists and researchers.

7. Support for Different Kernels: Jupyter Notebook supports different programming languages through kernels. You can install and use kernels for languages like R, Julia, and Scala, allowing you to work with multiple languages in the same notebook.

8. Reproducible Research: Jupyter Notebook promotes reproducibility in research by providing a complete record of code, text, and output in a single document. This makes it easier to share and reproduce scientific experiments and analysis.

Overall, Jupyter Notebook is a versatile tool that combines code execution, text, and visualizations in a single document. It provides an interactive and flexible environment for data exploration, analysis, and collaboration, making it a popular choice among data scientists, researchers, and educators.

5.5 VISUAL STUDIO CODE

Visual Studio Code (VS Code) is a free and lightweight source code editor developed by Microsoft. It is widely used by developers for various programming languages and platforms. Visual Studio Code (VS Code) can be used in several ways:

- Coding and Development: VS Code provides a powerful code editor with features like syntax highlighting, code completion, and IntelliSense, which facilitate writing ML algorithms for traffic prediction. You can write and debug your ML code directly in VS Code, making it convenient for implementing and testing different ML models.
- Python and ML Libraries: Python is a popular programming language for ML tasks, including traffic prediction. VS Code has excellent support for Python development, including a built-in Python interpreter and integration with popular ML libraries such as scikit-learn, TensorFlow, and PyTorch. You can leverage these libraries to build and train ML models for traffic prediction within the VS Code environment.
- Jupyter Notebooks: VS Code supports Jupyter Notebooks, which are interactive documents that allow you to combine code, visualizations, and explanatory text. Jupyter Notebooks are commonly used in ML tasks for data exploration, model development, and result analysis. You can create and work with Jupyter Notebooks in VS Code, making it easier to iterate on and document your traffic prediction experiments.
- Data Visualization: VS Code has extensions and integrations with data visualization libraries such as Matplotlib and Plotly, enabling you to create insightful visualizations of your traffic data. Visualizing the data can help you understand patterns, trends, and anomalies, which are crucial for developing accurate traffic prediction models.
- Git Integration and Collaboration: Traffic prediction projects often involve collaboration and version control. VS Code's built-in Git integration allows you to manage your code repository directly within the editor. You can easily commit, push, and pull changes, collaborate with team members, and track project history, ensuring smooth collaboration and code management.

- Terminal and Command-Line Tools: Traffic prediction projects may require running command-line tools or scripts for data preprocessing, model training, or evaluation. VS Code provides an integrated terminal, allowing you to execute command-line operations without leaving the editor. You can run scripts, manage dependencies, and interact with the commandline tools required for your traffic prediction ML workflow.
- Extension Ecosystem: VS Code has a vast extension ecosystem, including ML-specific extensions, that can enhance your traffic prediction workflow. These extensions provide additional functionality, such as data exploration tools, model evaluation metrics, automated hyperparameter tuning, and deployment options. You can explore and install relevant ML extensions from the VS Code Marketplace to augment your traffic prediction ML capabilities.

In summary, VS Code offers a flexible and feature-rich environment for developing and implementing ML models for traffic prediction. It provides coding support, integration with ML libraries, Jupyter Notebook capabilities, data visualization tools, collaboration features, command-line access, and a wide range of extensions to enhance your traffic prediction ML workflow.

5.6 ALGORITHM

5.6.**1 Random Forest**

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or the same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithms of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks. The following are the basic steps involved in performing the random forest algorithm

1. Pick N random records from the dataset.

2. Build a decision tree based on these N records.

3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

4. For the classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

5.6.2 Decision Tree

A Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

5.6.3 Support Vector Machine

Support Vector Machine is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N- dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane.
It becomes difficult to imagine when the number of features exceeds three.

5.6.4 Algorithm

1. Import the necessary libraries: \`warnings\`, \`pandas\`, \`numpy\`.

2. Set up warning filters to ignore warnings.

3. Import the required files.

4. Perform data preprocessing steps.

5. Perform feature extraction.

6. Create an Excel writer and write the DataFrame to a new sheet in the Excel file.

7. Close the Excel writer and save the Excel file.

8. Perform additional data preprocessing steps.

9. Perform data visualization and plot graphs.

10. Split the data into independent and dependent features.

11. Split the data into training and testing sets.

12. Implement the Random Forest classifier:

    - Create an instance of the Random Forest classifier.

    - Fit the model using the training set.

    - Make predictions on the test data.

    - Calculate the accuracy score.

    - Generate a confusion matrix.

13. Implement the Decision Tree classifier:

    - Create an instance of the Decision Tree classifier.

    - Fit the model using the training set.

    - Make predictions on the test data.

    - Calculate the accuracy score.

    - Generate a confusion matrix.

14. Implement the Support Vector Machine (SVM):

    - Create an instance of the SVM classifier.

    - Fit the model using the training set.

    - Make predictions on the test data.

    - Calculate the accuracy score.

    - Generate a confusion matrix.

15. Compare the accuracy of the three models using a bar chart.

16. Save the trained Random Forest model using 'pickle'.

17. Load the saved model using 'pickle'.

18. The code execution is complete.

5.7 PACKAGES

5.7.1 Pandas

- Pandas is a powerful library for data manipulation and analysis in Python.
- It provides data structures and functions to efficiently handle and process structured data.
  □ Pandas is widely used for tasks such as data cleaning, data exploration, data transformation, and data aggregation.
- It offers high-performance, easy-to-use data structures like DataFrames and Series, along with various data manipulation and analysis functions.

5.7.2 NumPy

- NumPy is a fundamental package for scientific computing in Python.

- It provides support for large, multi-dimensional arrays and matrices, along with a vast collection of mathematical functions to operate on these arrays.
- NumPy is highly efficient and optimized for numerical computations, making it a foundational package for tasks such as numerical analysis, linear algebra, statistics, and more.

5.7.3 **Scikit-learn (sklearn)**

Scikit-learn (sklearn) is a popular Python package for machine learning. It provides a wide range of tools and functionalities for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, and model selection. Scikitlearn is built on top of other scientific computing libraries in Python, such as NumPy and SciPy, and it is designed to be user-friendly and efficient.

Scikit-learn offers a unified and consistent API for implementing machine learning algorithms. It provides a wide range of algorithms and models, including support vector machines (SVM), random forests, decision trees, k-nearest neighbors, naive Bayes, and many more. Additionally, scikit-learn provides tools for data preprocessing, feature extraction, model evaluation, and performance metrics.

5.7.4 XlsxWriter

Xlswriter is a popular Python package for creating and manipulating Microsoft Excel files in the .xlsx format. The package is called "XlsxWriter," and it provides a straightforward and efficient way to generate Excel spreadsheets with various formatting options, formulas, charts, and more.With "XlsxWriter," you can create a new Excel file, add worksheets, write data to cells, apply formatting, merge cells, insert charts, and perform other operations to create complex Excel documents.

5.7.5 Seaborn

- Seaborn is a data visualization library built on top of Matplotlib.
- It provides a high-level interface for creating visually appealing statistical graphics.
- Seaborn simplifies the process of creating common statistical plots such as scatter plots, line plots, bar plots, histograms, and heatmaps.

- It offers enhanced aesthetics, built-in color palettes, and convenient functions for exploring relationships and patterns in data.

5.7.6 Matplotlib

- Matplotlib is a widely-used data visualization library in Python.
- It provides a comprehensive set of functions and classes for creating static, animated, and interactive visualizations.
- Matplotlib can be used to create various types of plots, including line plots, scatter plots, bar plots, histograms, pie charts, and more.
- It offers fine-grained control over plot elements and supports customization of colors, labels, annotations, and axes.

5.7.7 Pickle

- Pickle is a built-in Python module used for object serialization.
- It provides a way to convert Python objects into a byte stream that can be saved to a file or transferred over a network.
- Pickle allows you to store complex data structures, including lists, dictionaries, and custom objects, preserving their state.
- It is commonly used for tasks such as saving trained machine learning models to disk or transferring data between different Python environments.

5.7.8 Streamlit

- Streamlit is a Python package used for building interactive web applications for data science and machine learning.
- It simplifies the process of creating and sharing interactive dashboards, visualizations, and data exploration tools.
- Streamlit allows you to write simple Python scripts that automatically transform into web applications, without requiring extensive web development knowledge.

- It offers various features for creating interactive components, displaying data, and integrating with popular data analysis and machine learning libraries.

5.8 SOURCE CODE

```
"#importing require python packages\n",
   "import pandas as pd\n",
   "import numpy as np\n",
   "import json\n",
   "import seaborn as sns\n",
   "import matplotlib.pyplot as plt\n",
   "from IPython.display import display\n",
   "import plotly.express as px\n",
   "from collections import Counter\n",
   "from wordcloud import WordCloud\n",
   "from pyspark.sql import SparkSession #loading spark class\n",
   "from pyspark import SparkConf, SparkContext\n",
   "import webbrowser\n",
   "import os"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 3,
  "metadata": {},
  "outputs": [
   {
    "ename": "Exception",
    "evalue": "Java gateway process exited before sending its port number",
    "output_type": "error",
    "traceback": [
    ]
   }
  ],
   "source": [
    "spark = SparkSession.builder.appName(\"HDFS\").getOrCreate()\n",
    "sparkcont = SparkContext.getOrCreate(SparkConf().setAppName(\"HDFS\")) #creating spark object and initializing it\n",
    "logs = sparkcont.setLogLevel(\"ERROR\")\n",
    "filePath = os.path.abspath(\"Dataset/Youtube.csv\")\n",
    "df = spark.read.option(\"header\",\"true\").csv(\"file:///\"+filePath,inferSchema=True).limit(1000) #now loading dataset using spark\n",
    "dataset = df.toPandas()\n",
    "display(dataset)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "application/vnd.plotly.v1+json": {
       "config": {
        "plotlyServerURL": "https://plot.ly"
       },
       "data": [
        {
         "alignmentgroup": "True",
         "hovertemplate": "video_id=%{x}<br>views=%{marker.color}<extra></extra>",
         "legendgroup": "",
         "marker": {
          "color": [
   8072982,
   7797259,
   5792827,
   5541767,
   5085080,
   5029621,
   4972464,
   4722974,
   4716036,
   4477587,
   3621389,
   3447230,
   3300683,
   3058843,
   2935330,
   2918106,
   2812994,
   2682800
  ],
  "coloraxis": "coloraxis"
 },
 "name": "",
 "offsetgroup": "",
 "orientation": "v",
 "showlegend": false,
 "textposition": "auto",
 "type": "bar",
 "x": [
  "p8XP7A7kvzM",
  "coOKvrsmQiI",
    "f-UzOpuKOVY",
    "6p-QzY5bxJ0",
"JzCsM1vtn78",
    "f-UzOpuKOVY",
    "JzCsM1vtn78",        "p8XP7A7kvzM",
    "nRafaCcfrcI",
    "JzCsM1vtn78",
"sottGW1p5os",
    "bAkEd8r7Nnw",
    "bAkEd8r7Nnw",
"8hKbIhrb1WU",        "nRafaCcfrcI",
    "bAkEd8r7Nnw",
        "fpxBxp9QKrk",
     "8hKbIhrb1WU"
   ],
   "xaxis": "x",
   "y": [
    8072982,
    7797259,
    5792827,
    5541767,
    5085080,
    5029621,
    4972464,
    4722974,
    4716036,
    4477587,
    3621389,
    3447230,
    3300683,
    3058843,
    2935330,
    2918106,
    2812994,
    2682800
   ],
   "yaxis": "y"
  }
 ],
 "layout": {
  "barmode": "relative",
  "coloraxis": {
   "colorbar": {
    "title": {
     "text": "views"
    }
   },
   "colorscale": [
    [
     0,
     "#440154"
    ],
    [
     0.1111111111111111,
     "#482878"
    ],
    [
     0.2222222222222222,
     "#3e4989"
    ],
  [
   0.3333333333333333,
   "#31688e"
  ],
  [
   0.4444444444444444,
   "#26828e"
  ],
  [
   0.5555555555555556,
   "#1f9e89"
  ],
  [
   0.6666666666666666,
   "#35b779"
  ],
  [
   0.7777777777777778,
   "#6ece58"
  ],
  [
   0.8888888888888888,
   "#b5de2b"
  ],
  [
   1,
   "#fde725"
  ]
 ]
},
"legend": {
 "tracegroupgap": 0
},
"template": {
 "data": {
  "bar": [
   {
    "error_x": {
     "color": "#2a3f5f"
    },
    "error_y": {
     "color": "#2a3f5f"
    },
    "marker": {
     "line": {
      "color": "#E5ECF6",
      "width": 0.5
     }
    },
    "type": "bar"
   }
  ],
  "barpolar": [
   {
    "marker": {
     "line": {
      "color": "#E5ECF6",
      "width": 0.5
     }
    },
  "type": "barpolar"
 }
],
"carpet": [
 {
  "aaxis": {
   "endlinecolor": "#2a3f5f",
   "gridcolor": "white",
   "linecolor": "white",
   "minorgridcolor": "white",
   "startlinecolor": "#2a3f5f"
  },
  "baxis": {
   "endlinecolor": "#2a3f5f",
   "gridcolor": "white",           "linecolor": "white",
   "minorgridcolor": "white",
   "startlinecolor": "#2a3f5f"
  },
  "type": "carpet"
 }
],
"choropleth": [
 {
  "colorbar": {
   "outlinewidth": 0,
   "ticks": ""
  },
  "type": "choropleth"
 }
],
"contour": [
 {
  "colorbar": {
   "outlinewidth": 0,
   "ticks": ""
  },
  "colorscale": [
   [
    0,
    "#0d0887"
   ],
   [
    0.1111111111111111,
    "#46039f"
   ],
   [
    0.2222222222222222,              "#7201a8"
   ],
  "backgroundcolor": "#E5ECF6",
  "gridcolor": "white",
  "gridwidth": 2,
  "linecolor": "white",
  "showbackground": true,
  "ticks": "",
  "zerolinecolor": "white"
 },
 "zaxis": {
  "backgroundcolor": "#E5ECF6",
  "gridcolor": "white",
  "gridwidth": 2,
  "linecolor": "white",
  "showbackground": true,
  "ticks": "",
  "zerolinecolor": "white"
 }
},
"shapedefaults": {
 "line": {
  "color": "#2a3f5f"
 }
},
"ternary": {
 "aaxis": {
  "gridcolor": "white",
  "linecolor": "white",
  "ticks": ""
 },
 "baxis": {
  "gridcolor": "white",
  "linecolor": "white",
  "ticks": ""
 },
 "bgcolor": "#E5ECF6",
 "caxis": {
  "gridcolor": "white",
  "linecolor": "white",
  "ticks": ""
 }
},
"title": {
   "x": 0.05
  },
  "xaxis": {
   "automargin": true,
   "gridcolor": "white",
   "linecolor": "white",
   "ticks": "",
   "title": {
    "standoff": 15
   },
   "zerolinecolor": "white",
   "zerolinewidth": 2
  },
  "yaxis": {
   "automargin": true,
   "gridcolor": "white",
   "linecolor": "white",
   "ticks": "",
   "title": {
    "standoff": 15
   },
   "zerolinecolor": "white",            "zerolinewidth": 2
  }
 }
},
"title": {
 "text": "Top 10 Videos Watched"
},
"xaxis": {
 "anchor": "y",
      "domain": [
       0,
       1
      ],
      "title": {
       "text": "video_id"
      }
     },
     "yaxis": {
      "anchor": "x",
      "domain": [
       0,
       1
      ],
      "title": {
       "text": "views"
      }
     }
    }
   },
   ]
  },
  "metadata": {},
  "output_type": "display_data"
 }
],
"source": [
 "#finding top 10 watched videos\n",
 "dataset = pd.read_csv(\"Dataset/Youtube.csv\", nrows=1000)\n",
   "videos = dataset.query(\"category_id == 24\").sort_values(by=['views'],
ascending=False)\n",
   "videos = videos[0:18]\n",
   "fig = px.bar(videos,\n",
   "             x='video_id',\n",
   "             y='views',\n",
   "             color='views',\n",
   "             color_continuous_scale='Viridis',\n",
   "             title='Top 10 Videos Watched')\n",
   "fig.show()"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 4,
  "metadata": {},
  "outputs": [
   {
    "data": {
     "application/vnd.plotly.v1+json": {
      "config": {
       "plotlyServerURL": "https://plot.ly"
      },
      "data": [
       {
        "alignmentgroup": "True",
        "hovertemplate": "video_id=%{x}<br>views=%{marker.color}<extra></extra>",
"legendgroup": "",
        "marker": {
         "color": [
.8888888888888888,
            "#fdca26"
           ],
           [
            1,
            "#f0f921"
           ]
          ],
          "type": "histogram2d"
         }
        ],
        "histogram2dcontour": [
         {
          "colorbar": {
           "outlinewidth": 0,
           "ticks": ""
          },
          "colorscale": [
           [
            0,
            "#0d0887"
           ],
           [
            0.1111111111111111,
            "#46039f"
           ],
           [
            0.2222222222222222,
            "#7201a8"
           ],
 [
  0.3333333333333333,
  "#9c179e"
 ],
 [
  0.4444444444444444,
  "#bd3786"
 ],
 [
  0.5555555555555556,
  "#d8576b"
 ],
 [
  0.6666666666666666,
  "#ed7953"
 ],
 [
  0.7777777777777778,
  "#fb9f3a"
 ],
 [
  0.8888888888888888,
  "#fdca26"
 ],
 [
  1,
  "#f0f921"
 ]
],
"type": "histogram2dcontour"
```

```
 }
],
"mesh3d": [
 {
  "colorbar": {
   "outlinewidth": 0,
   "ticks": ""
  },
  "type": "mesh3d"
 }
],
"parcoords": [
 {
  "line": {
   "colorbar": {
    "outlinewidth": 0,
    "ticks": ""
   }
  },
  "type": "parcoords"
 }
],
"pie": [
 {
  "automargin": true,
  "type": "pie"
 }
],
"scatter": [
 {
```

```
"marker": {
 "colorbar": {
  "outlinewidth": 0,
  "ticks": ""
 }
},
"type": "scatter"
```

```
}
],
"scatter3d": [
 {
  "line": {
   "colorbar": {
    "outlinewidth": 0,
    "ticks": ""
   }
  },
  "marker": {
   "colorbar": {
    "outlinewidth": 0,
    "ticks": ""
   }
  },
  "type": "scatter3d"
 }
],
"scattercarpet": [
 {
  "marker": {
   "colorbar": {
    "outlinewidth": 0,
    "ticks": ""
   }


 {
  },
  "type": "scattercarpet"
 }
],
"scattergeo": [

  "marker": {
   "colorbar": {
    "outlinewidth": 0,
    "ticks": ""
   }
  },
  "type": "scattergeo"
 }
],
"scattergl": [
 {
  "marker": {
   "colorbar": {
    "outlinewidth": 0,
    "ticks": ""
   }
  },
  "type": "scattergl"
 }


 {
```

```
],
"scattermapbox": [
 {
  "marker": {
   "colorbar": {
    "outlinewidth": 0,
    "ticks": ""
   }
  },
  "type": "scattermapbox"
 }
],
"scatterpolar": [

  "marker": {
   "colorbar": {
    "outlinewidth": 0,
    "ticks": ""
   }
  },
  "type": "scatterpolar"
 }
],
"scatterpolargl": [
 {


 {
  "marker": {
   "colorbar": {
    "outlinewidth": 0,
    "ticks": ""
   }
  },
  "type": "scatterpolargl"
 }
],
"scatterternary": [
 {
  "marker": {
   "colorbar": {
    "outlinewidth": 0,
    "ticks": ""
   }
  },
  "type": "scatterternary"
 }
],
"surface": [

  "colorbar": {




 {
  "outlinewidth": 0,
  "ticks": ""
 },
 "colorscale": [
  [
   0,
   "#0d0887"
  ],
  [
   0.1111111111111111,
   "#46039f"
  ],
  [
   0.2222222222222222,
   "#7201a8"
  ],
  [
   0.3333333333333333,
   "#9c179e"
  ],
  [
   0.4444444444444444,
   "#bd3786"
  ],
  [
   0.5555555555555556,
   "#d8576b"
   ],
   [
    0.6666666666666666,
    "#ed7953"
   ],
   [
    0.7777777777777778,
    "#fb9f3a"
   ],
   [
    0.8888888888888888,
    "#fdca26"
   ],
   [
    1,
    "#f0f921"
   ]
  ],
  "type": "surface"
 }
],
"table": [
 {
  "cells": {
   "fill": {
    "color": "#EBF0F8"
   },
   "line": {
    "color": "white"
   }
```

```
    },
    "header": {
     "fill": {
      "color": "#C8D4E3"
     },
     "line": {
      "color": "white"
     }
    },
    "type": "table"
   }
  ]
 },
 "layout": {
  "annotationdefaults": {
   "arrowcolor": "#2a3f5f",
   "arrowhead": 0,
   "arrowwidth": 1
  },
  "coloraxis": {
   "colorbar": {
    "outlinewidth": 0,
    "ticks": ""
   }
  },
  "colorscale": {
   "diverging": [
    [
     0,
 "#8e0152"
],
[
 0.1,
 "#c51b7d"
],
[
 0.2,
 "#de77ae"
],
[
 0.3,
 "#f1b6da"
],
[
 0.4,
 "#fde0ef"
],
[
 0.5,
 "#f7f7f7"
],
[
 0.6,
 "#e6f5d0"
],
[
 0.7,
 "#b8e186"
],
 [
  0.8,
  "#7fbc41"
 ],
 [
  0.9,
  "#4d9221"
 ],
 [
  1,
  "#276419"
 ]
],
"sequential": [
 [
  0,
  "#0d0887"
 ],
 [
  0.1111111111111111,
  "#46039f"
 ],
 [
  0.2222222222222222,
  "#7201a8"
 ],
 [
  0.3333333333333333,
  "#9c179e"
  ],
  [
   0.4444444444444444,
   "#bd3786"
  ],
  [
   0.5555555555555556,
   "#d8576b"
  ],
  [
   0.6666666666666666,
   "#ed7953"
  ],
  [
   0.7777777777777778,
   "#fb9f3a"
  ],
  [
   0.8888888888888888, "#fdca26"

  ],
  [
   1,
   "#f0f921"
  ]
 ],
 "sequentialminus": [
  [
   0,
   "#0d0887"
```

```
],
[
 0.1111111111111111,
 "#46039f"
],
[
 0.2222222222222222,
 "#7201a8"
],
[
 0.3333333333333333,
 "#9c179e"
],
[
 0.4444444444444444,
 "#bd3786"
],
[
 0.5555555555555556,
 "#d8576b"
],
[
 0.6666666666666666,
               "#ed7953"
],
[
 0.7777777777777778,
 "#fb9f3a"
],
    [
     0.8888888888888888,
     "#fdca26"
    ],
    [
     1,
     "#f0f921"
    ]
   ]
  },
  "colorway": [
   "#636efa",
   "#EF553B",
   "#00cc96",
   "#ab63fa",
   "#FFA15A",
   "#19d3f3",
   "#FF6692",
   "#B6E880",
"#FF97FF",
   "#FECB52"
  ],
  "font": {
   "color": "#2a3f5f"
  },
  "geo": {
   "bgcolor": "white",          "lakecolor": "white",
   "landcolor": "#E5ECF6",
   "showlakes": true,
   "showland": true,
 "subunitcolor": "white"
},
"hoverlabel": {
 "align": "left"
},
"hovermode": "closest",
"mapbox": {
 "style": "light"
},
"paper_bgcolor": "white",
"plot_bgcolor": "#E5ECF6",
"polar": {
 "angularaxis": {
  "gridcolor": "white",
  "linecolor": "white",
  "ticks": ""
 },
 "bgcolor": "#E5ECF6",
 "radialaxis": {
  "gridcolor": "white",
  "linecolor": "white",
  "ticks": ""
 }
},
"scene": {
 "xaxis": {
  "backgroundcolor": "#E5ECF6",
  "gridcolor": "white",
  "gridwidth": 2,
   "linecolor": "white",
   "showbackground": true,
   "ticks": "",
   "zerolinecolor": "white"
  },
  "yaxis": {
   "backgroundcolor": "#E5ECF6",
   "gridcolor": "white",
   "gridwidth": 2,
   "linecolor": "white",
   "showbackground": true,
   "ticks": "",
   "zerolinecolor": "white"
  },
  "zaxis": {
   "backgroundcolor": "#E5ECF6",
   "gridcolor": "white",
   "gridwidth": 2,
   "linecolor": "white",
   "showbackground": true,
   "ticks": "",
   "zerolinecolor": "white"
  }
 },
 "shapedefaults": {
  "line": {
   "color": "#2a3f5f"
  }
 },
 "ternary": {
```

```
 "aaxis": {
  "gridcolor": "white",
  "linecolor": "white",
  "ticks": ""
 },
 "baxis": {
  "gridcolor": "white",
  "linecolor": "white",
  "ticks": ""
 },
 "bgcolor": "#E5ECF6",
 "caxis": {
  "gridcolor": "white",
  "linecolor": "white",
  "ticks": ""
 }
},
"title": {
 "x": 0.05
},
"xaxis": {
 "automargin": true,
 "gridcolor": "white",
 "linecolor": "white",
 "ticks": "",
 "title": {
  "standoff": 15
 },
```

CHAPTER 6
TESTING

6.1 TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. Software Testing can also be stated as the process of validating and verifying that a software program/application/product:

- Meets the business and technical requirements that guided its design and Development.
- Works as expected and can be implemented with the same characteristics.

6.2 TESTING METHODS

6.2.1 Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

- Functions: Identified functions must be exercised.
- Output: Identified classes of software outputs must be exercised.
- Systems/Procedures: system should work properly

6.2.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

6.3 VALIDATION

Here in machine learning we are dealing with dataset which is in excel sheet format so if any test case we need means we need to check excel file. Later on classification will work on the respective columns of dataset.

Test Case 1:

TEST CASE NAME	DESCRIPTION	STEP NO	ACTION TO BE TAKEN(DESIGN STEPS)	EXPECTED (DESIGN STEP)	TEST EXECUTION RESULT
Excel sheet verification	Objective: There should be an Excel sheet. Any number of rows can be added to the Excel sheet.	Step 1	Excel sheet should be available	The Excel sheet is available	Pass
		Step 2	Excel sheet is created based on the template	The Excel sheet should always be based on the template	Pass
		Step 3	Changed the name of the Excel sheet	Should not make any modification on the name of excel sheet	Fail
		Step 4	Added 10000 or above records	Can add any number of records	Pass

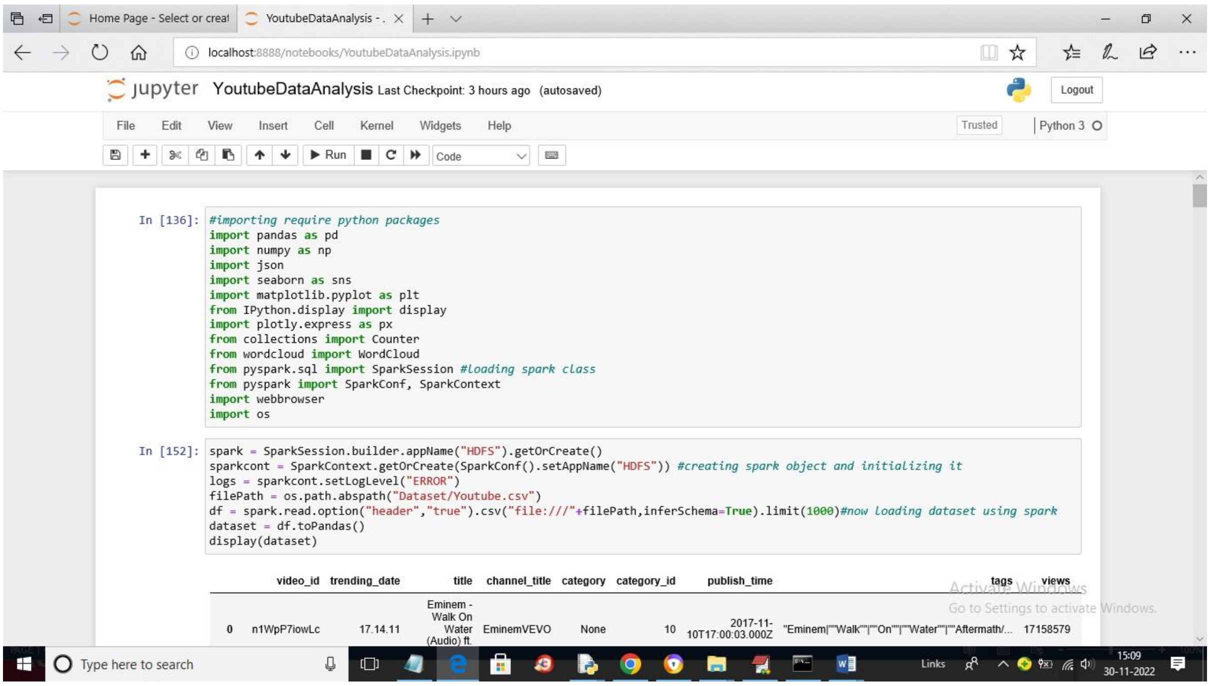
Table 6.3.1: Test case validation.

CHAPTER 7
RESULTS

7.1 SCREENSHOTS

YouTube Data Analysis using Hadoop and Spark

In this project we are using YouTube dataset of more than 250 MB for various analysis such as finding top watch videos, top trending videos, top videos upload in each category. To process huge data we are using SPARK packages and below are the output screen. This project consists of so many graphs so we coded using JUPYTER notebook



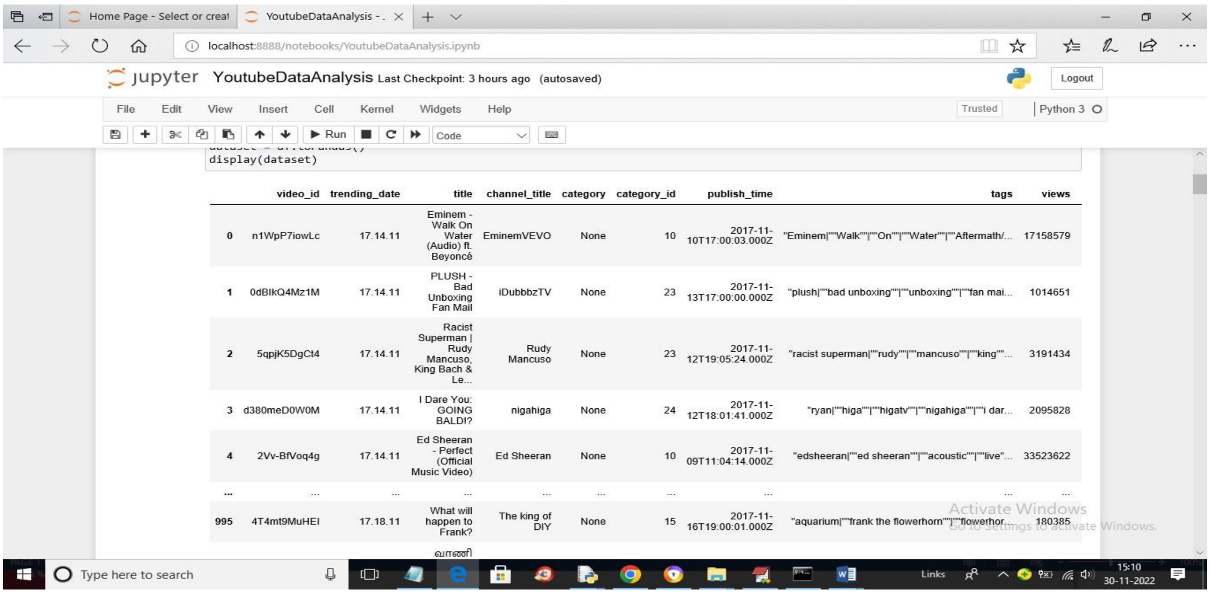
Home Page - Select or creat
YoutubeDataAnalysis - .
localhost:8888/notebooks/YoutubeDataAnalysis.ipynb
jupyter YoutubeDataAnalysis Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 ○
Run
Code
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
15:09
30-11-2022

```
In [136]: #importing require python packages
          import pandas as pd
          import numpy as np
          import json
          import seaborn as sns
          import matplotlib.pyplot as plt
          from IPython.display import display
          import plotly.express as px
          from collections import Counter
          from wordcloud import WordCloud
          from pyspark.sql import SparkSession #loading spark class
          from pyspark import SparkConf, SparkContext
          import webbrowser
          import os

In [152]: spark = SparkSession.builder.appName("HDFS").getOrCreate()
          sparkcont = SparkContext.getOrCreate(SparkConf().setAppName("HDFS")) #creating spark object and initializing it
          logs = sparkcont.setLogLevel("ERROR")
          filePath = os.path.abspath("Dataset/Youtube.csv")
          df = spark.read.option("header","true").csv("file:///"+filePath,inferSchema=True).limit(1000)#now loading dataset using spark
          dataset = df.toPandas()
          display(dataset)
```

```
     video_id  trending_date        title  channel_title  category  category_id         publish_time                                             tags      views
                                 Eminem -
                                  Walk On
0  n1WpP7iowLc         17.14.11     Water  EminemVEVO      None             10            2017-11-  "Eminem|""Walk""|""On""|""Water""|""Aftermath/...  17158579
                                (Audio) ft.                                       10T17:00:03.000Z
```

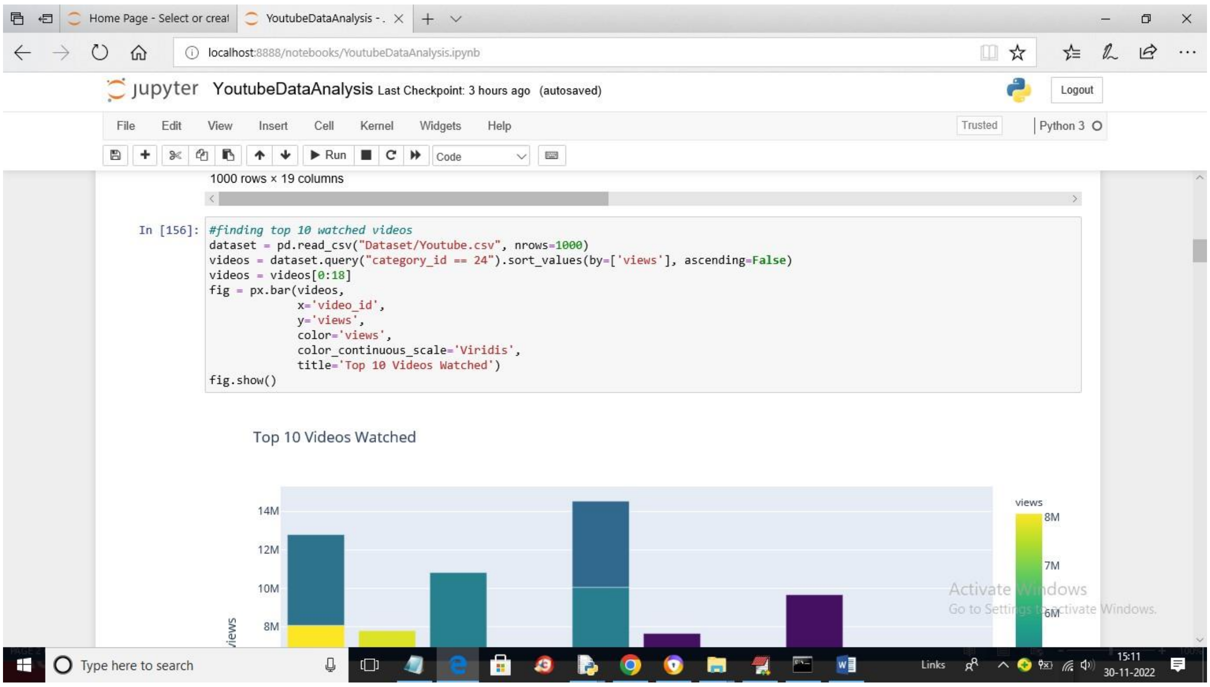
7.1.1 In above screen we are importing require python packages and then using SPARK class we are loading dataset and then in below screen we are displaying loaded dataset values



```
display(dataset)
```

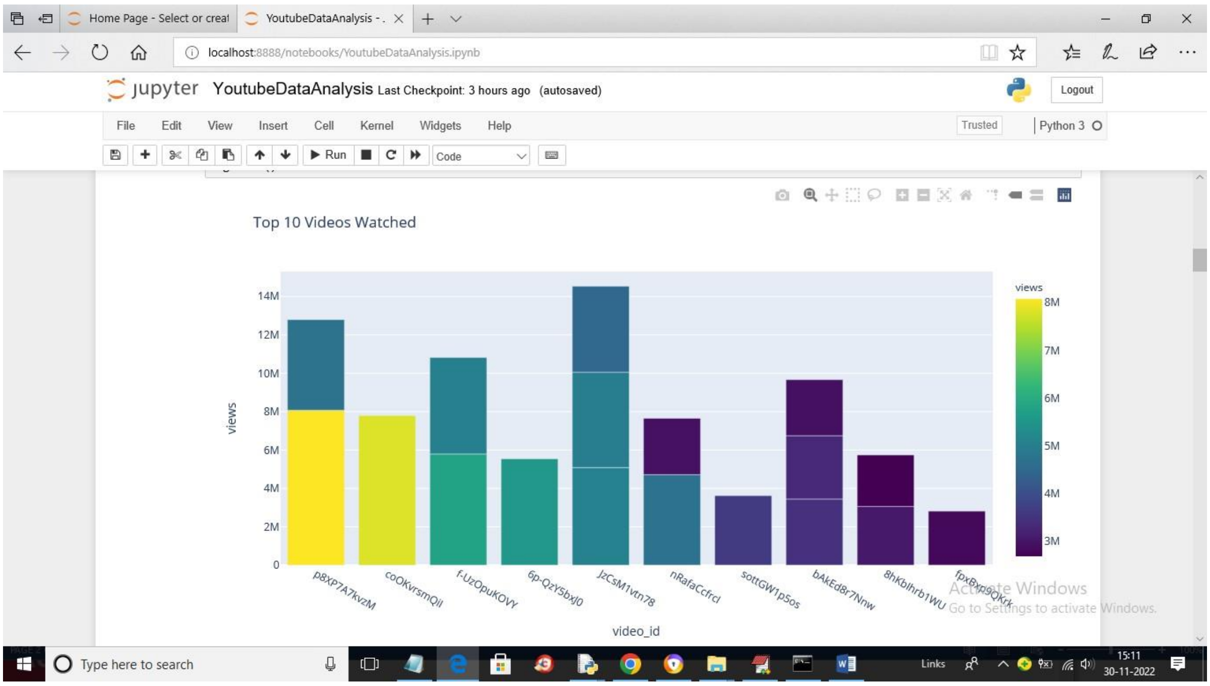
	video_id	trending_date	title	channel_title	category	category_id	publish_time	tags	views
0	n1WpP7iowLc	17.14.11	Eminem - Walk On Water (Audio) ft. Beyoncé	EminemVEVO	None	10	2017-11-10T17:00:03.000Z	"Eminem|""Walk""|""On""|""Water""|""Aftermath/...	17158579
1	0dBIkQ4Mz1M	17.14.11	PLUSH - Bad Unboxing Fan Mail	iDubbbzTV	None	23	2017-11-13T17:00:00.000Z	"plush|""bad unboxing""|""unboxing""|""fan mai...	1014651
2	5qpjK5DgCt4	17.14.11	Racist Superman | Rudy Mancuso, King Bach & Le...	Rudy Mancuso	None	23	2017-11-12T19:05:24.000Z	"racist superman|""rudy""|""mancuso""|""king""...	3191434
3	d380meD0W0M	17.14.11	I Dare You: GOING BALD!?	nigahiga	None	24	2017-11-12T18:01:41.000Z	"ryan|""higa""|""higatv""|""nigahiga""|""i dar...	2095828
4	2Vv-BfVoq4g	17.14.11	Ed Sheeran - Perfect (Official Music Video)	Ed Sheeran	None	10	2017-11-09T11:04:14.000Z	"edsheeran|""ed sheeran""|""acoustic""|""live"...	33523622
...	...	...	...	...	...	...	...	...	...
995	4T4mt9MuHEI	17.18.11	What will happen to Frank?	The king of DIY	None	15	2017-11-16T19:00:01.000Z	"aquarium|""frank the flowerhorn""|""flowerhor...	180385
			வாணி						

7.1.2 In above screen dataset loaded and displaying

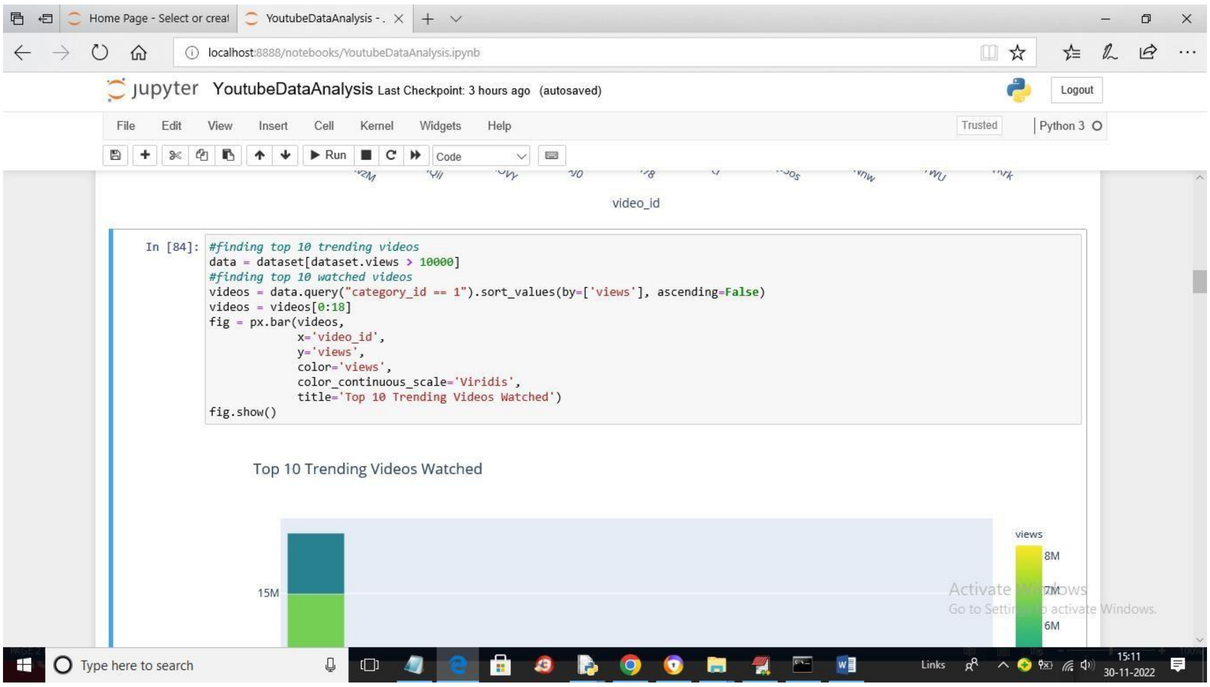


1000 rows × 19 columns

```
In [156]: #finding top 10 watched videos
dataset = pd.read_csv("Dataset/Youtube.csv", nrows=1000)
videos = dataset.query("category_id == 24").sort_values(by=['views'], ascending=False)
videos = videos[0:18]
fig = px.bar(videos,
             x='video_id',
             y='views',
             color='views',
             color_continuous_scale='Viridis',
             title='Top 10 Videos Watched')
fig.show()
```

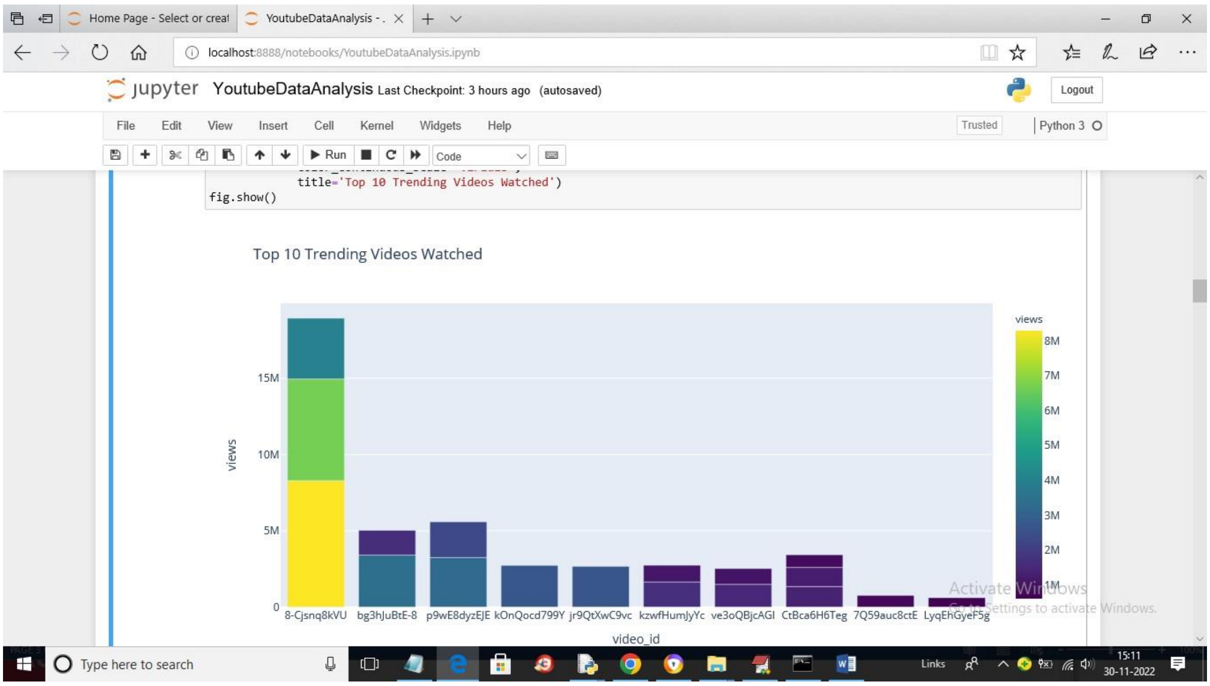


7.1.3 In above 2 screen we are showing code and output of top 10 watched videos where x-axis contains video ID and y-axis represents COUNT


Home Page - Select or creat
YoutubeDataAnalysis - .
localhost:8888/notebooks/YoutubeDataAnalysis.ipynb
jupyter YoutubeDataAnalysis Last Checkpoint: 3 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
video_id

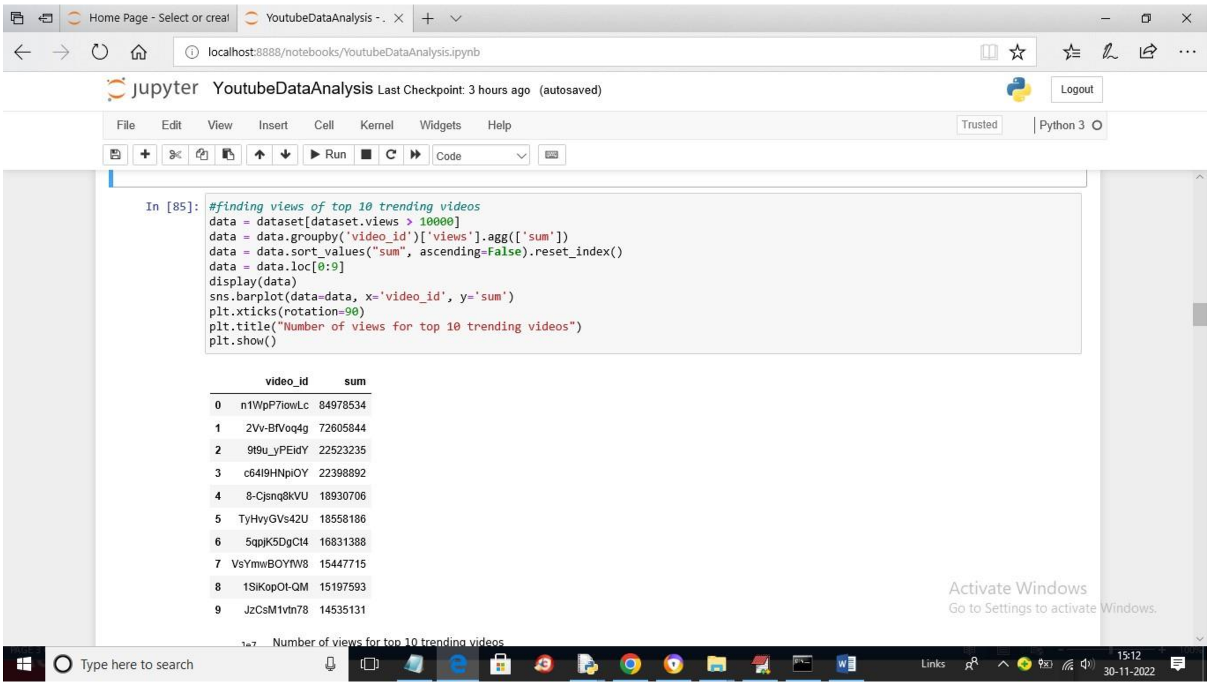
```
In [84]: #finding top 10 trending videos
         data = dataset[dataset.views > 10000]
         #finding top 10 watched videos
         videos = data.query("category_id == 1").sort_values(by=['views'], ascending=False)
         videos = videos[0:18]
         fig = px.bar(videos,
                      x='video_id',
                      y='views',
                      color='views',
                      color_continuous_scale='Viridis',
                      title='Top 10 Trending Videos Watched')
         fig.show()
```

Top 10 Trending Videos Watched
15M
views
8M
7M
6M
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
15:11
30-11-2022



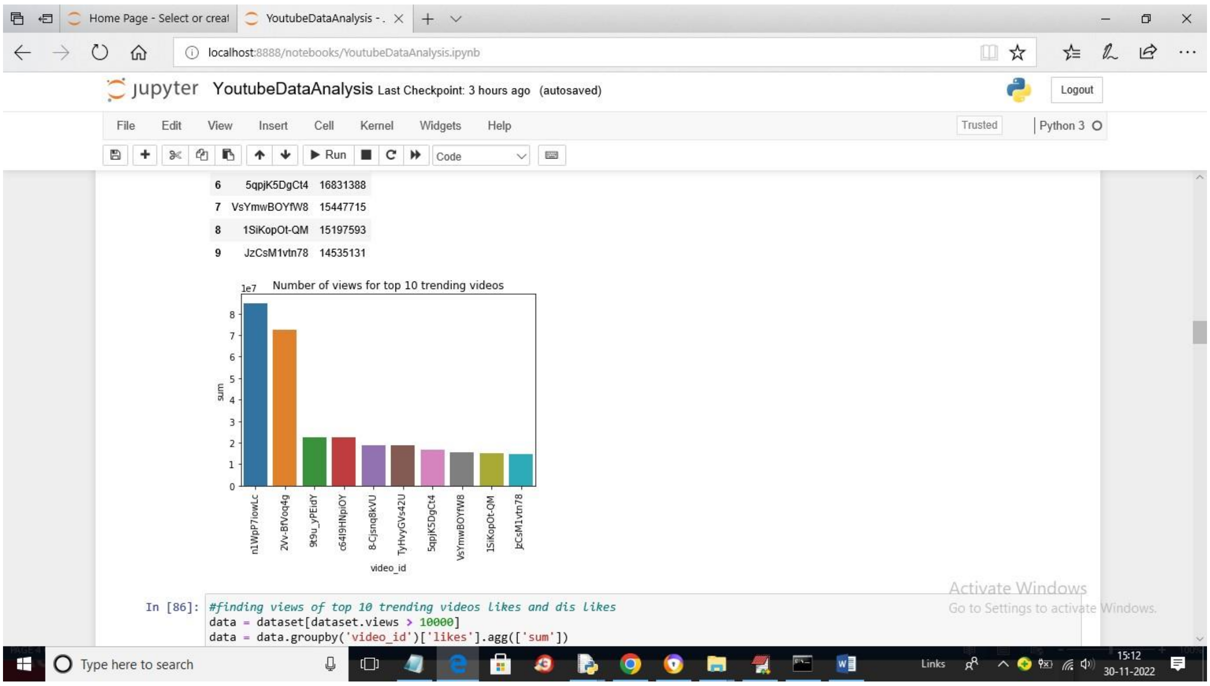
```
        title='Top 10 Trending Videos Watched')
fig.show()
```

7.1.4 In above screen we are showing TOP 10 trending videos


Home Page - Select or creat
YoutubeDataAnalysis - .
localhost:8888/notebooks/YoutubeDataAnalysis.ipynb
jupyter
YoutubeDataAnalysis Last Checkpoint: 3 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
1e7
Number of views for top 10 trending videos
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
15:12
30-11-2022

```
In [85]: #finding views of top 10 trending videos
         data = dataset[dataset.views > 10000]
         data = data.groupby('video_id')['views'].agg(['sum'])
         data = data.sort_values("sum", ascending=False).reset_index()
         data = data.loc[0:9]
         display(data)
         sns.barplot(data=data, x='video_id', y='sum')
         plt.xticks(rotation=90)
         plt.title("Number of views for top 10 trending videos")
         plt.show()
```

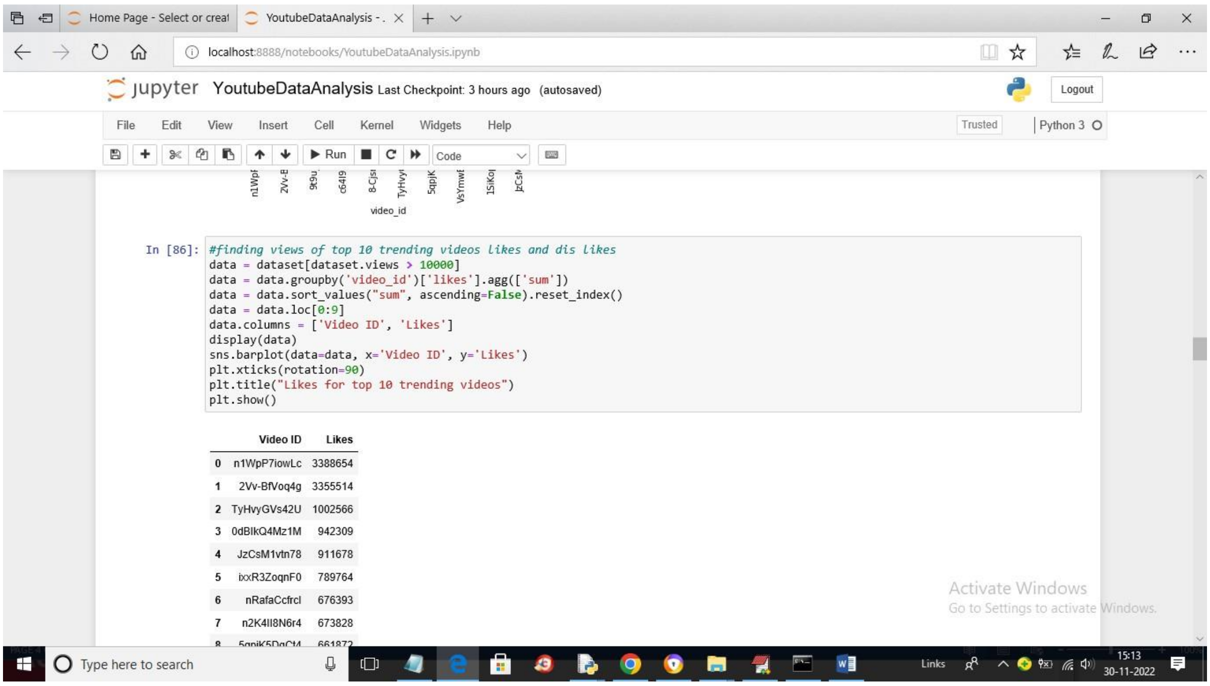
	video_id	sum
0	n1WpP7iowLc	84978534
1	2Vv-BfVoq4g	72605844
2	9t9u_yPEidY	22523235
3	c64I9HNpiOY	22398892
4	8-Cjsnq8kVU	18930706
5	TyHvyGVs42U	18558186
6	5qpjK5DgCt4	16831388
7	VsYmwBOYfW8	15447715
8	1SiKopOt-QM	15197593
9	JzCsM1vtn78	14535131



6	5qpjK5DgCt4	16831388
7	VsYmwBOYfW8	15447715
8	1SiKopOt-QM	15197593
9	JzCsM1vtn78	14535131

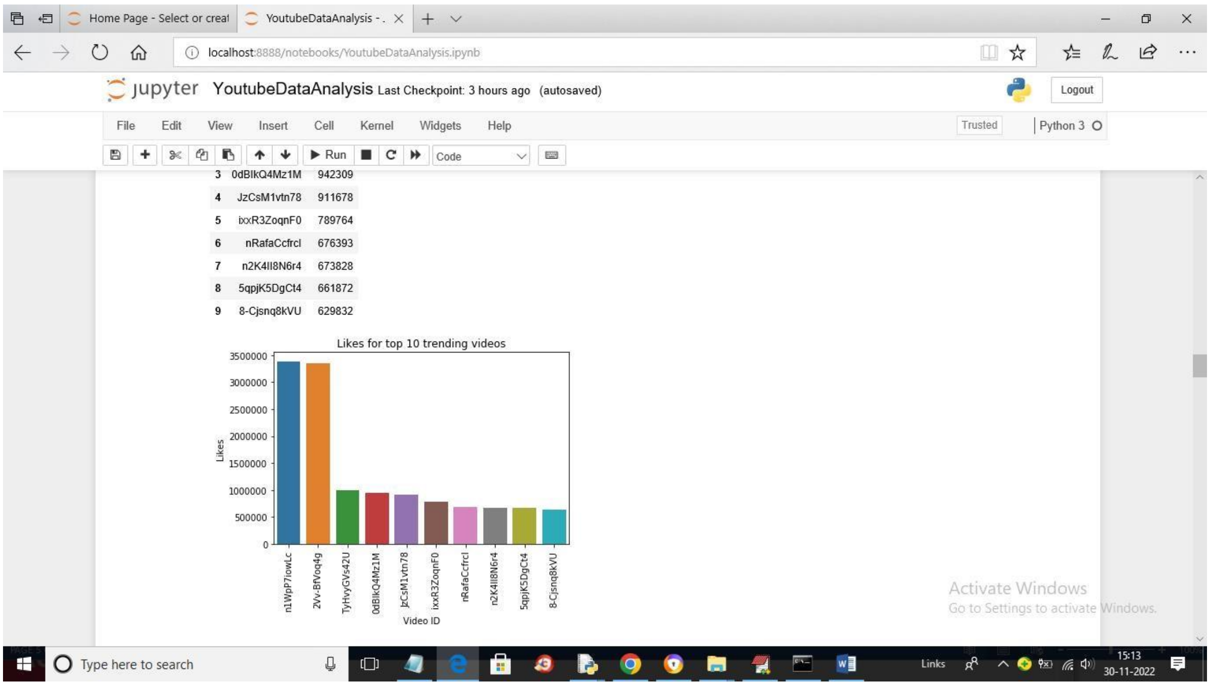
```
In [86]: #finding views of top 10 trending videos likes and dis likes
         data = dataset[dataset.views > 10000]
         data = data.groupby('video_id')['likes'].agg(['sum'])
```

7.1.5 In above two screens we are finding views of top 10 trending videos



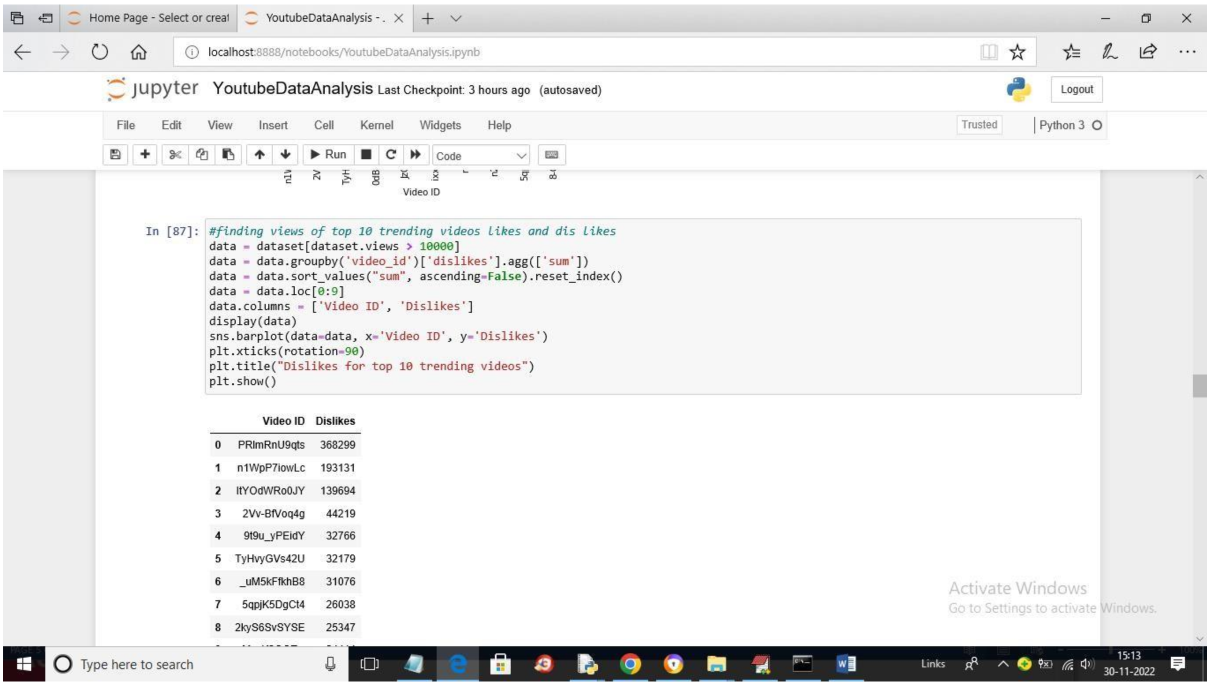
```
In [86]: #finding views of top 10 trending videos likes and dis likes
         data = dataset[dataset.views > 10000]
         data = data.groupby('video_id')['likes'].agg(['sum'])
         data = data.sort_values("sum", ascending=False).reset_index()
         data = data.loc[0:9]
         data.columns = ['Video ID', 'Likes']
         display(data)
         sns.barplot(data=data, x='Video ID', y='Likes')
         plt.xticks(rotation=90)
         plt.title("Likes for top 10 trending videos")
         plt.show()
```

	Video ID	Likes
0	n1WpP7iowLc	3388654
1	2Vv-BfVoq4g	3355514
2	TyHvyGVs42U	1002566
3	0dBIkQ4Mz1M	942309
4	JzCsM1vtn78	911678
5	ixxR3ZoqnF0	789764
6	nRafaCcfrcl	676393
7	n2K4II8N6r4	673828
8	5qpjK5DgCt4	661872



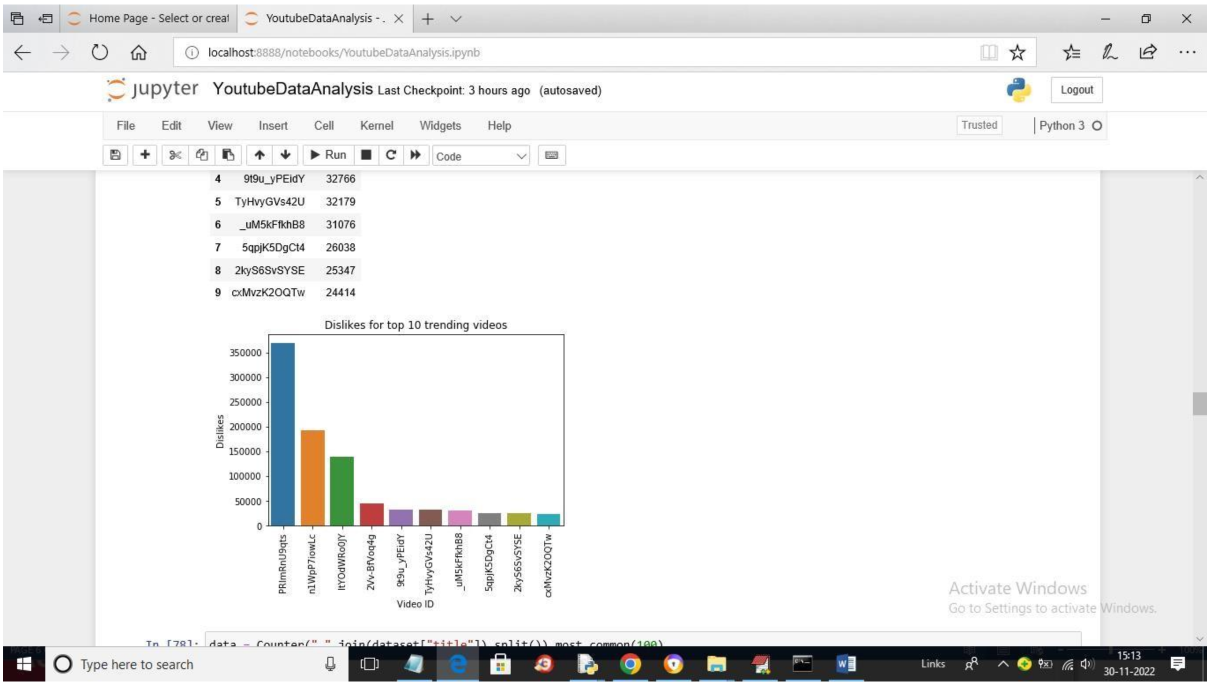
3	0dBIkQ4Mz1M	942309
4	JzCsM1vtn78	911678
5	ixxR3ZoqnF0	789764
6	nRafaCcfrcl	676393
7	n2K4II8N6r4	673828
8	5qpjK5DgCt4	661872
9	8-Cjsnq8kVU	629832

7.1.6 In above screen we are showing graph and code for top 10 trending Videos LIKES


Home Page - Select or creat
YoutubeDataAnalysis - .
localhost:8888/notebooks/YoutubeDataAnalysis.ipynb
jupyter
YoutubeDataAnalysis Last Checkpoint: 3 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Video ID
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
15:13
30-11-2022

```
In [87]: #finding views of top 10 trending videos likes and dis likes
         data = dataset[dataset.views > 10000]
         data = data.groupby('video_id')['dislikes'].agg(['sum'])
         data = data.sort_values("sum", ascending=False).reset_index()
         data = data.loc[0:9]
         data.columns = ['Video ID', 'Dislikes']
         display(data)
         sns.barplot(data=data, x='Video ID', y='Dislikes')
         plt.xticks(rotation=90)
         plt.title("Dislikes for top 10 trending videos")
         plt.show()
```

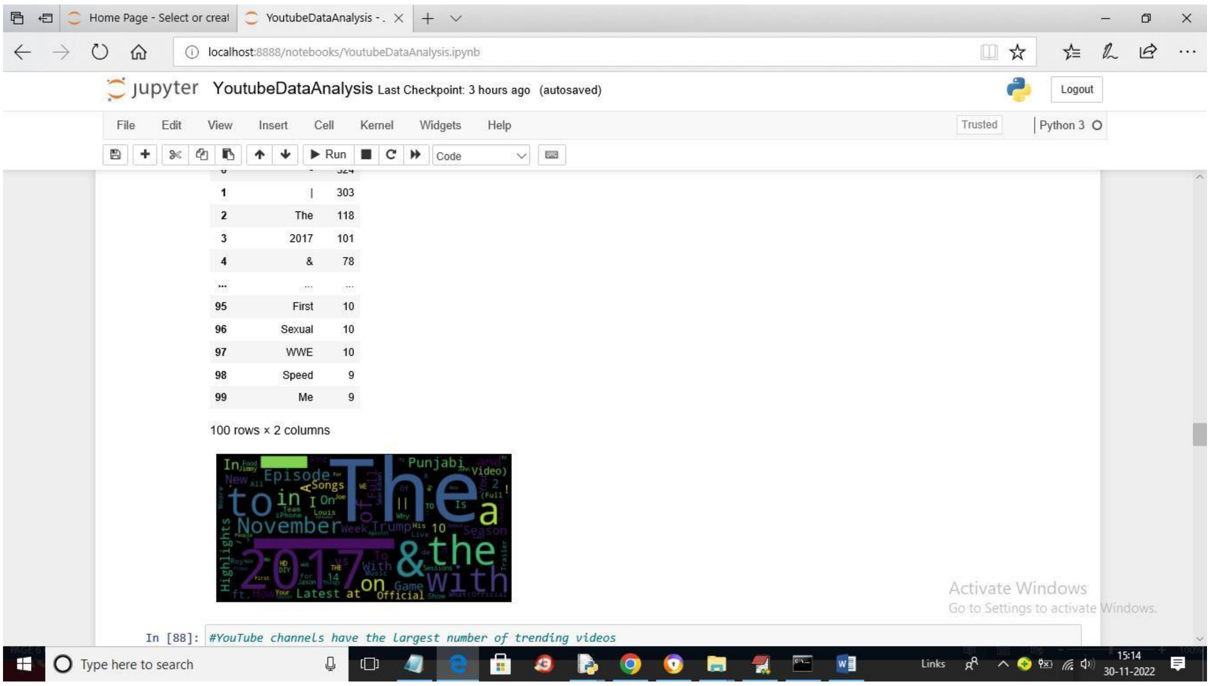
	Video ID	Dislikes
0	PRlmRnU9qts	368299
1	n1WpP7iowLc	193131
2	ltYOdWRo0JY	139694
3	2Vv-BfVoq4g	44219
4	9t9u_yPEidY	32766
5	TyHvyGVs42U	32179
6	_uM5kFfkhB8	31076
7	5qpjK5DgCt4	26038
8	2kyS6SvSYSE	25347



```
4   9t9u_yPEidY   32766
5   TyHvyGVs42U   32179
6   _uM5kFfkhB8   31076
7   5qpjK5DgCt4   26038
8   2kyS6SvSYSE   25347
9   cxMvzK2OQTw   24414
```

```
In [78]: data = Counter(" ".join(dataset["title"]).split()).most_common(100)
```

7.1.7 In above screen we are plotting DISLIKES graph for trending videos

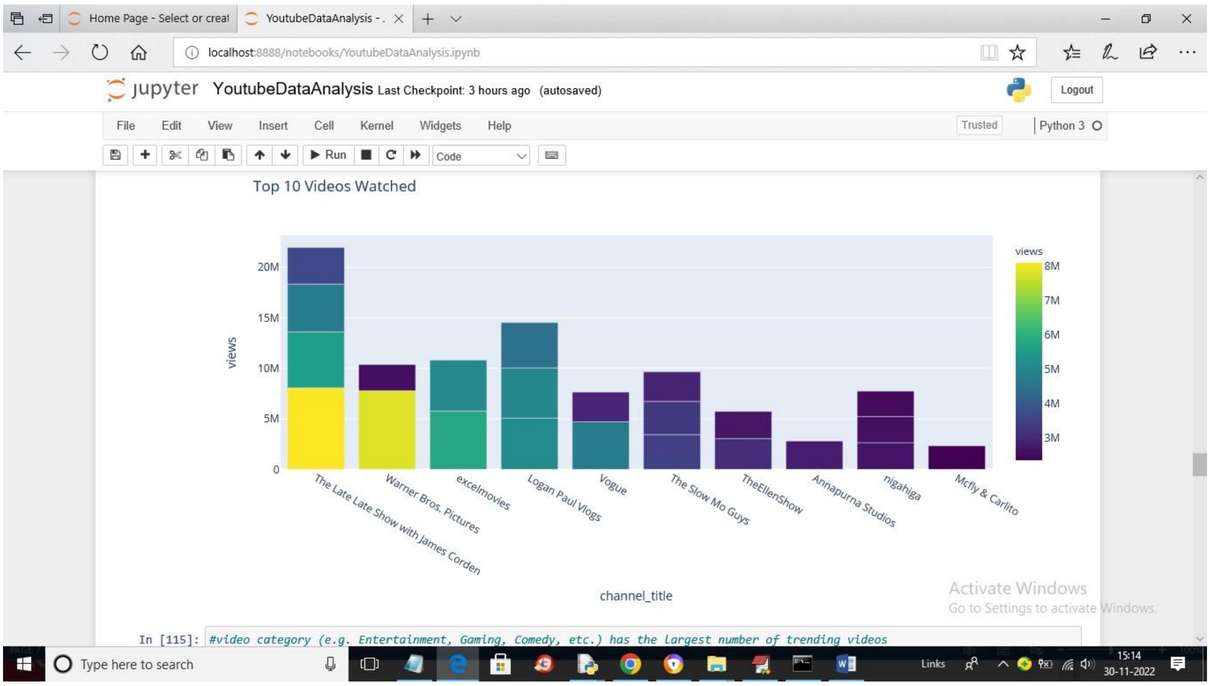


```
1        |   303
2      The   118
3     2017   101
4        &    78
...    ...   ...
95   First    10
96  Sexual    10
97     WWE    10
98   Speed     9
99      Me     9

100 rows × 2 columns
```

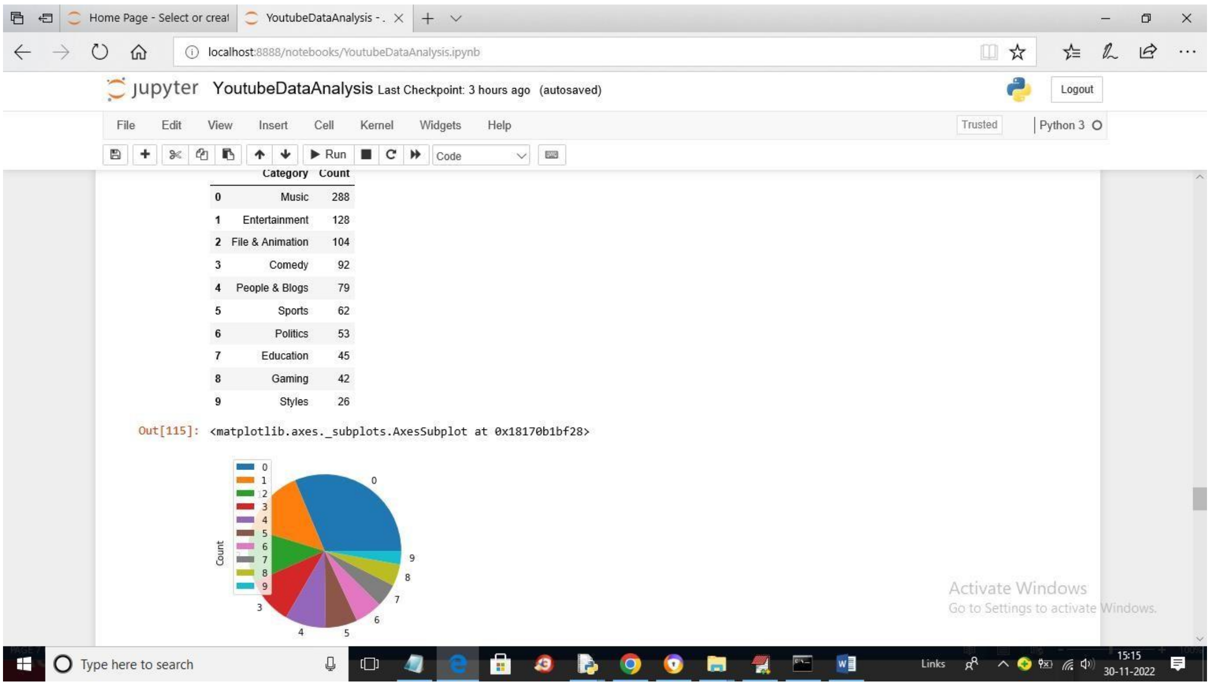
```
In [88]: #YouTube channels have the largest number of trending videos
```

7.1.8 In above screen we are plotting and displaying count of most common words



```
In [115]: #video category (e.g. Entertainment, Gaming, Comedy, etc.) has the largest number of trending videos
```

7.1.9 In above graph we are showing TOP 10 channels watched

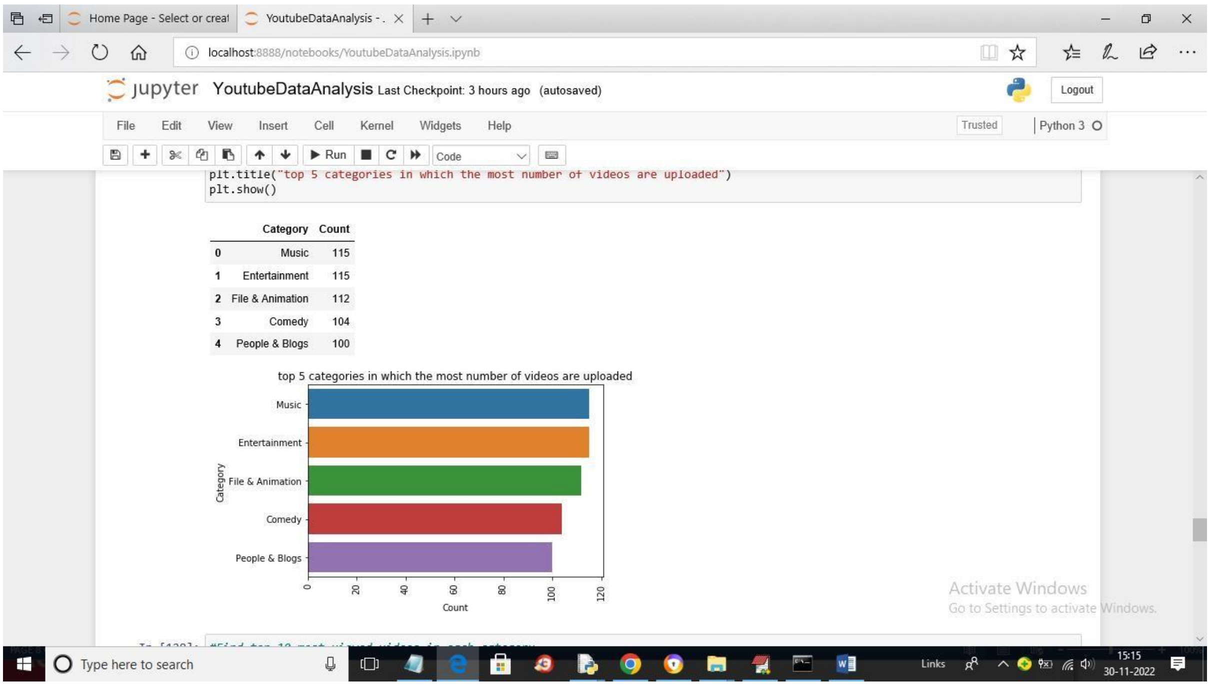

Home Page - Select or creat
YoutubeDataAnalysis - .
localhost:8888/notebooks/YoutubeDataAnalysis.ipynb
jupyter YoutubeDataAnalysis Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

	Category	Count
0	Music	288
1	Entertainment	128
2	File & Animation	104
3	Comedy	92
4	People & Blogs	79
5	Sports	62
6	Politics	53
7	Education	45
8	Gaming	42
9	Styles	26

```
Out[115]: <matplotlib.axes._subplots.AxesSubplot at 0x18170b1bf28>
```

0
1
2
3
4
5
6
7
8
9
Count
0
9
8
7
6
5
4
3
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
15:15
30-11-2022

In above graph we are showing largest trending videos based on category

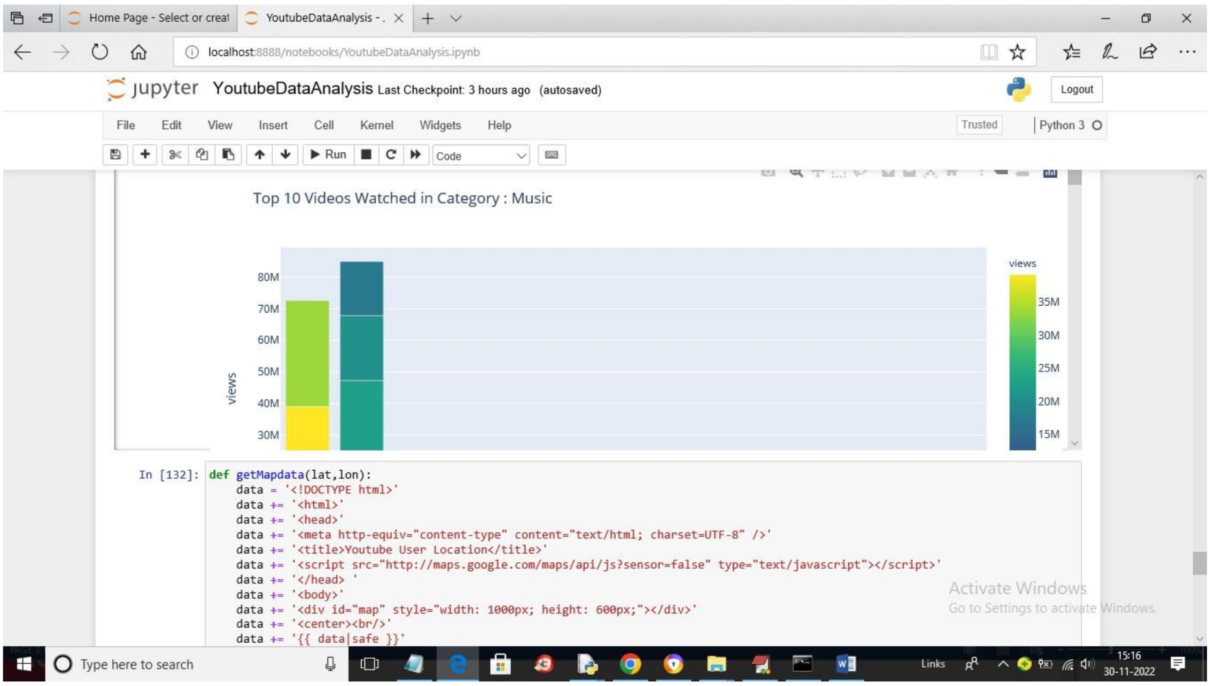


```
plt.title("top 5 categories in which the most number of videos are uploaded")
plt.show()
```

	Category	Count
0	Music	115
1	Entertainment	115
2	File & Animation	112
3	Comedy	104
4	People & Blogs	100

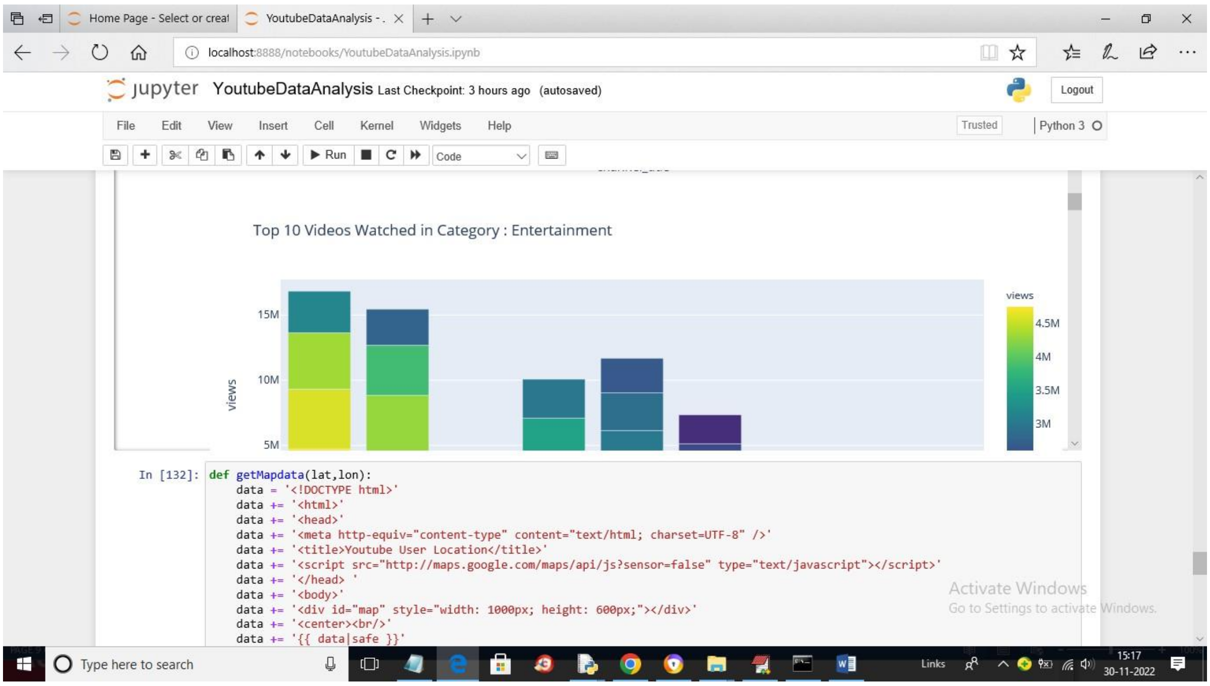
top 5 categories in which the most number of videos are uploaded

7.1.10 In above screen showing top 5 categories videos uploaded


Home Page - Select or creat
YoutubeDataAnalysis - .
localhost:8888/notebooks/YoutubeDataAnalysis.ipynb
jupyter
YoutubeDataAnalysis Last Checkpoint: 3 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Top 10 Videos Watched in Category : Music
views
80M
70M
60M
50M
40M
30M
views
35M
30M
25M
20M
15M
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
15:16
30-11-2022

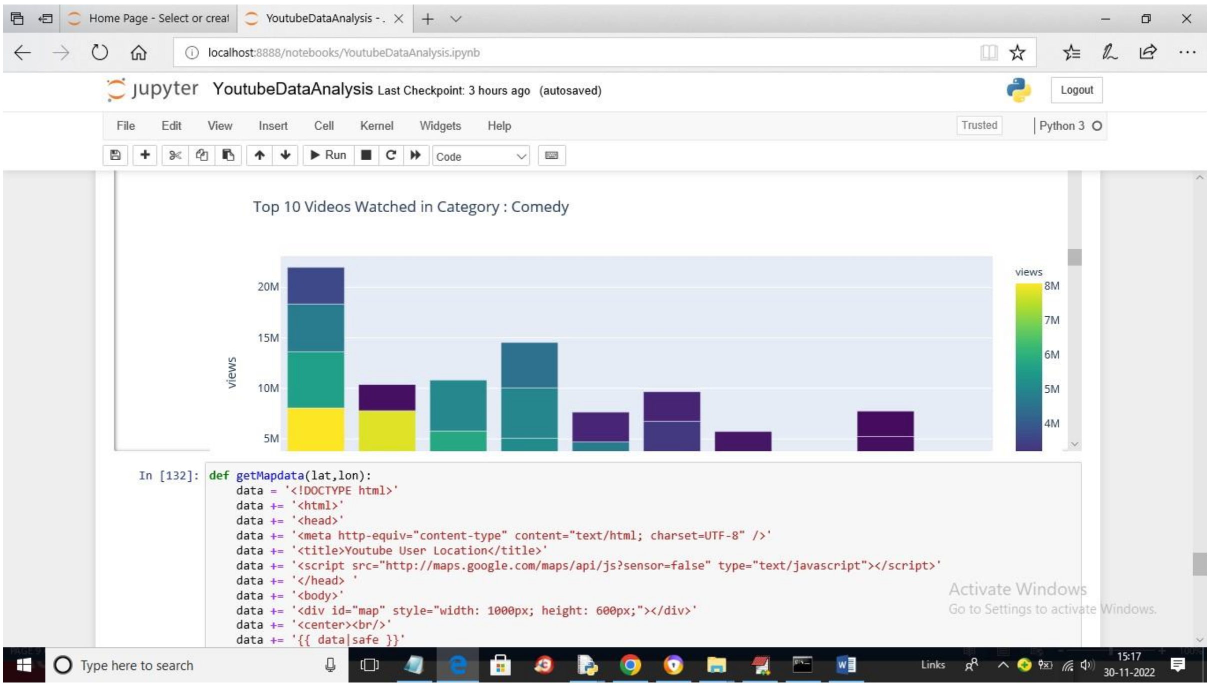
```
In [132]: def getMapdata(lat,lon):
              data = '<!DOCTYPE html>'
              data += '<html>'
              data += '<head>'
              data += '<meta http-equiv="content-type" content="text/html; charset=UTF-8" />'
              data += '<title>Youtube User Location</title>'
              data += '<script src="http://maps.google.com/maps/api/js?sensor=false" type="text/javascript"></script>'
              data += '</head> '
              data += '<body>'
              data += '<div id="map" style="width: 1000px; height: 600px;"></div>'
              data += '<center><br/>'
              data += '{{ data|safe }}'
```

7.1.11 In above graph showing Top 10 videos watch in music category


Home Page - Select or creat
YoutubeDataAnalysis -
localhost:8888/notebooks/YoutubeDataAnalysis.ipynb
jupyter YoutubeDataAnalysis Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Top 10 Videos Watched in Category : Entertainment
views
15M
10M
5M
views
4.5M
4M
3.5M
3M
Type here to search
Links
15:17
30-11-2022

```
In [132]: def getMapdata(lat,lon):
              data = '<!DOCTYPE html>'
              data += '<html>'
              data += '<head>'
              data += '<meta http-equiv="content-type" content="text/html; charset=UTF-8" />'
              data += '<title>Youtube User Location</title>'
              data += '<script src="http://maps.google.com/maps/api/js?sensor=false" type="text/javascript"></script>'
              data += '</head> '
              data += '<body>'
              data += '<div id="map" style="width: 1000px; height: 600px;"></div>'
              data += '<center><br/>'
              data += '{{ data|safe }}'
```

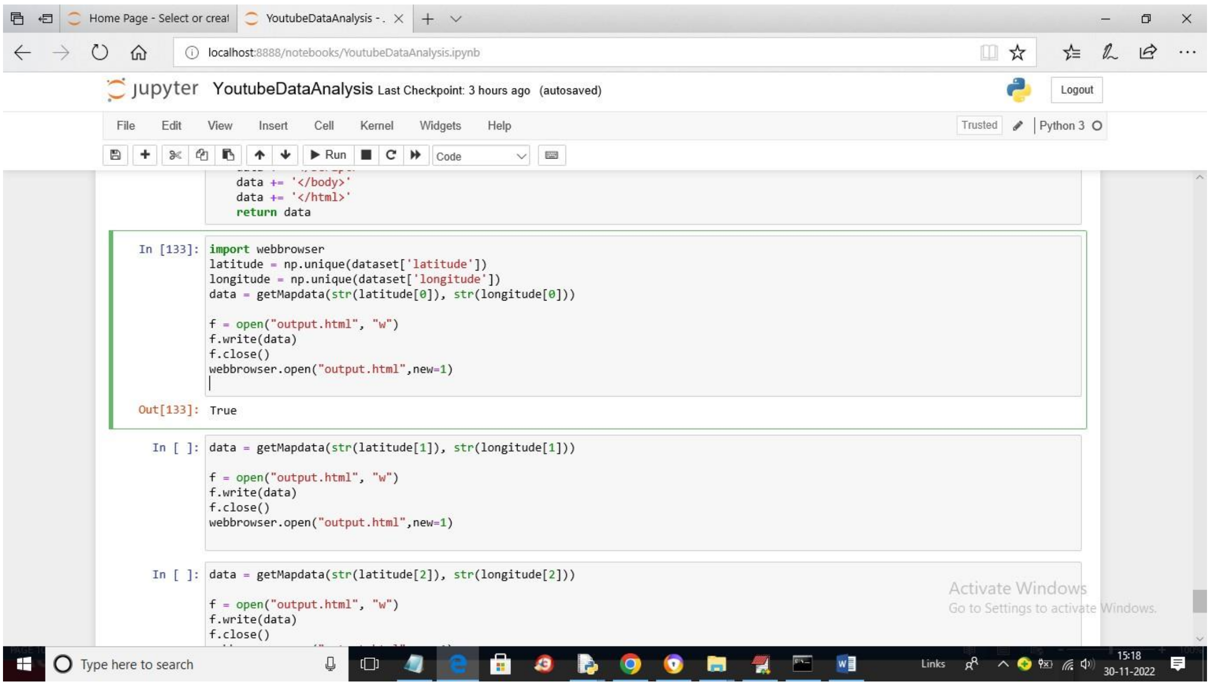
7.1.12 Above graph for Entertainment category


Home Page - Select or creat
YoutubeDataAnalysis -
localhost:8888/notebooks/YoutubeDataAnalysis.ipynb
jupyter YoutubeDataAnalysis Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Top 10 Videos Watched in Category : Comedy
views
20M
15M
10M
5M
views
8M
7M
6M
5M
4M
Type here to search
Links
15:17
30-11-2022

```
In [132]: def getMapdata(lat,lon):
              data = '<!DOCTYPE html>'
              data += '<html>'
              data += '<head>'
              data += '<meta http-equiv="content-type" content="text/html; charset=UTF-8" />'
              data += '<title>Youtube User Location</title>'
              data += '<script src="http://maps.google.com/maps/api/js?sensor=false" type="text/javascript"></script>'
              data += '</head> '
              data += '<body>'
              data += '<div id="map" style="width: 1000px; height: 600px;"></div>'
              data += '<center><br/>'
              data += '{{ data|safe }}'
```

7.1.13 Above graph for comedy category and similarly you can see graph for each category

In below screen run block to view user location in MAP


Home Page - Select or creat
YoutubeDataAnalysis - .
localhost:8888/notebooks/YoutubeDataAnalysis.ipynb
jupyter YoutubeDataAnalysis Last Checkpoint: 3 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
15:18
30-11-2022

```
    data += '</body>'
    data += '</html>'
    return data
```

```
In [133]: import webbrowser
          latitude = np.unique(dataset['latitude'])
          longitude = np.unique(dataset['longitude'])
          data = getMapdata(str(latitude[0]), str(longitude[0]))

          f = open("output.html", "w")
          f.write(data)
          f.close()
          webbrowser.open("output.html",new=1)

Out[133]: True
```

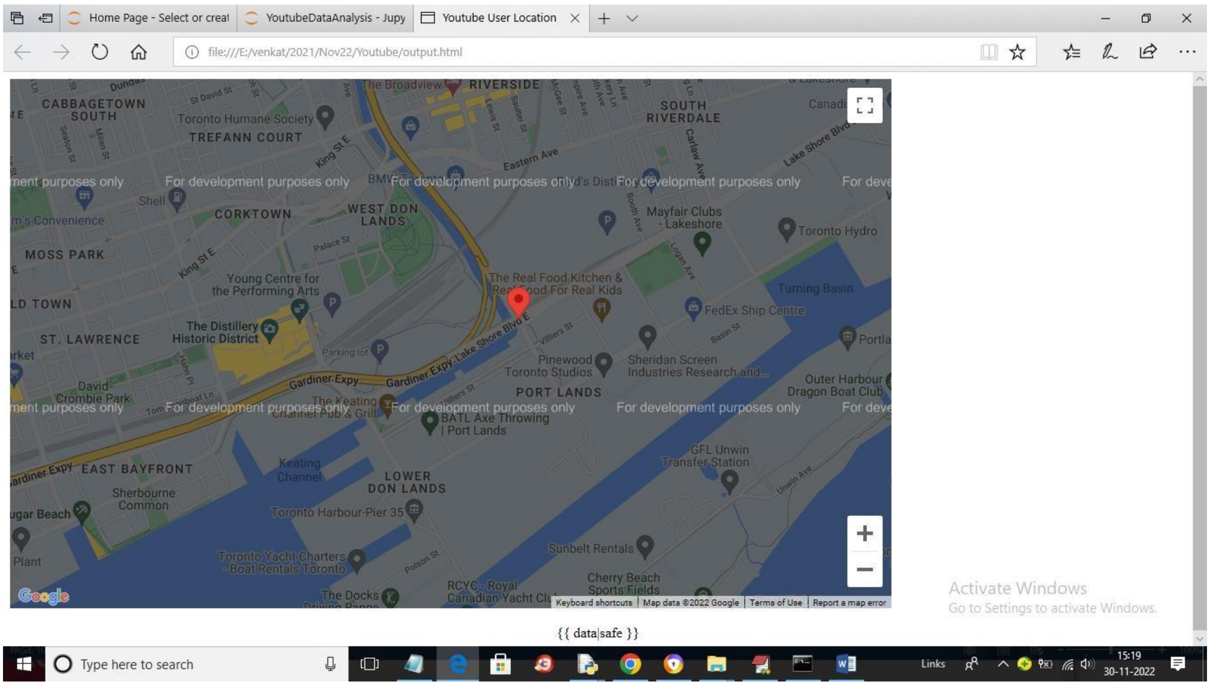
```
In [ ]: data = getMapdata(str(latitude[1]), str(longitude[1]))

        f = open("output.html", "w")
        f.write(data)
        f.close()
        webbrowser.open("output.html",new=1)
```

```
In [ ]: data = getMapdata(str(latitude[2]), str(longitude[2]))

        f = open("output.html", "w")
        f.write(data)
        f.close()
```

7.1.14 In above screen run each block to view user location MAP like below screen


Home Page - Select or creat
YoutubeDataAnalysis - Jupy
Youtube User Location
file:///E:/venkat/2021/Nov22/Youtube/output.html
CABBAGETOWN SOUTH
Toronto Humane Society
TREFANN COURT
RIVERSIDE
SOUTH RIVERDALE
For development purposes only
CORKTOWN
WEST DON LANDS
Mayfair Clubs - Lakeshore
Toronto Hydro
MOSS PARK
Young Centre for the Performing Arts
The Real Food Kitchen & Real Food For Real Kids
Turning Basin
FedEx Ship Centre
ST. LAWRENCE
The Distillery Historic District
Pinewood Toronto Studios
Sheridan Screen Industries Research and...
Outer Harbour Dragon Boat Club
PORT LANDS
Gardiner Expy
Lake Shore Blvd E
BATL Axe Throwing | Port Lands
EAST BAYFRONT
Sherbourne Common
Keating Channel
LOWER DON LANDS
GFL Unwin Transfer Station
Toronto Harbour-Pier 35
Toronto Yacht Charters - Boat Rentals Toronto
Sunbelt Rentals
Cherry Beach Sports Fields
RCYC - Royal Canadian Yacht Cl
The Docks
Google
Keyboard shortcuts | Map data ©2022 Google | Terms of Use | Report a map error
{{ data|safe }}
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
15:19
30-11-2022

CHAPTER 8

CONCLUSION

8.1 CONCLUSION

Main aim of the study is to identify the trending pattern of YouTube video games using sentiment analysis. Study present the method for increase the accuracy in the sentiment analysis and predict the trending game videos in YouTube based on their features and classification analysis such as SVM, Naive Bayes, Logistic Regression and Random Forest. In our module, we achieved 80% accuracy in Naive Bayes and SVM classification. The study identified the best algorithm and Classification Model in this research. Moreover, it is recommended the best trending videos patterns based on the YouTube channels metadata. The research results have provided the recommendations for the YouTube video features such as video publish time, video's title length, views and likes. The study will provide the benefit for the YouTubers who wanted to popularize the video content among the viewers. In the further research, we need to take increase the accuracy by testing other classifications and improve the more prediction accuracy of the YouTube game videos and evaluating the outcome for the different type of video channels.

CHAPTER 9

FUTURE ENHANCEMENTS

9.1 Technology:

1. **Artificial Intelligence and Machine Learning:**
    - Enhanced AI capabilities for more sophisticated tasks, improved natural language understanding, and reasoning.
    - Advancements in unsupervised learning, reinforcement learning, and neural network architectures.
2. **Quantum Computing:**
    - Developing scalable and stable quantum computers to solve complex problems efficiently.
    - Progress in quantum algorithms and error correction.
3. **Biotechnology and Health Sciences:**
    - Personalized medicine using genomics, gene editing (CRISPR), and targeted therapies.
    - Advancements in regenerative medicine and organ/tissue engineering.
4. **Renewable Energy:**
    - Breakthroughs in energy storage, solar, wind, and other renewable sources. • Improving efficiency and reducing costs in renewable energy technologies.
5. **Space Exploration:**
    - Human missions to Mars and deeper exploration of outer space.
    - Advancements in satellite technology for communication, Earth observation, and exploration.

REFERENCES

1. Youngsub Han ; Kwangmi Ko Kim, “Sentiment Analysis on Social Media Using Morphological Sentence Pattern Model,” 7-9 June 2017 [2017 IEEE 15 th International Conference on Software Engineering Research, Management and Applications (SERA)]
2. Chen CP, “Exploring personal branding on YouTube”,(2013), Journel Internet Commer, 12:332-347.
3. C. Li,” Characterizing and Predicting the Popularity of Online Videos”,(2016), EEE Access, 1630-1641.
4. Pozzi FA, Fersini E, Messina E, Liu B (2016),”Sentiement analysis in social networks,” 1at edn, Morgan Kaufmann Publishers.
5. N. Irtiza Tripto and M. Eunus Ali, "Detecting Multilabel Sentiment and Emotions from Bangla YouTube Comments," 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), Sylhet, 2018, pp. 1-6, doi: 10.1109/ICBSLP.2018.8554875.
6. G. Mohana Prabha, B. Madhumitha, R. P. Ramya, “Predicting the Popularity of Trending Videos in Youtube Using Sentimental Analysis”,(2019) International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue6S3,
7. Soleymani, Mohammad, et al. "A survey of multimodal sentiment analysis." Image and Vision Computing 65 (2017): 3-14.
8. Rangaswamy, Shanta, et al. "Metadata extraction and classification of youtube films the usage of sentiment evaluation." IEEE International Carnahan Conference on Security Technology (ICCST). 2016.
9. Mulholland, Eleanor, et al. "Analysing Emotional Sentiment in People’s YouTube Channel Comments." Interactivity, Game Creation, Design, Learning, and Innovation. Springer, Cham, 2016. 181-188.

10. S. Chelaru, C. Orellana-Rodriguez and I. S. Altingovde, “Howuseful is social feedback for learning to rank YouTube videos?” In World Wide Web, 17(5), 2013, pp. 1-29.
11. Hanif Bhuiyan; Jinat Ara; Rajon Bardhan; Md. Rashedul Islam, “Retrieving YouTube video by sentiment analysis on user comment,” 2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)
12. F. Pedregosa, in press, “Scikit-learn: Machine Learning in Python,” In: Journal of Machine Learning Research 12 (2011), pp. 2825-2830.
13. Stefan Siersdorfer, in press. “How Useful Are Your Comments? Analyzing and Predicting Youtube Comments and Comment Ratings”. In: Proceedings of the 19th International Conference on World Wide Web. WWW ’10. Raleigh, North Carolina, USA: ACM, 2010, pp. 891-900. ISBN: 978-1-60558-799-8. DOI: 10.1145/1772690. 1772781.
14. S. Rangaswamy, S. Ghosh, and S. Jha, “Metadata Extraction Analysis,” 24-27 Oct. 2016 [2016 IEEE International Carnahan Conference on Security Technology (ICCST)]
15. Wei-Lun Chang, “Will Sentiments in Comments Influence Online Video Popularity?” 2018 IEEE International Conference on Big Data (Big Data)
16. Y. Song, M. Zhao, J. Yagnik, and X. Wu. “Taxonomic classification for web-based videos.,” In Proc. IEEE Conf. Computer Vision and Pattern Recognition, June 2010.
17. Lakshmish Kaushik; Abhijeet Sangwan; John H. L. Hansen, “Automatic sentiment extraction from YouTube videos,” 2013 IEEE Workshop on Automatic Speech Recognition and Understanding [8-12 Dec. 2013].