Tutorial 1

Installing Ubuntu in VirtualBox

Introduction

This tutorial covers how to install Ubuntu Linux in VirtualBox on Windows, Mac, or Linux. Ubuntu is a popular Linux distribution that is easy to use for beginners. VirtualBox allows you to install Ubuntu in a virtual machine so you don't have to disturb your main operating system.

Downloading Ubuntu and VirtualBox

First download the Ubuntu ISO image from the Ubuntu website. The recommended version is the latest Long Term Support (LTS) release. LTS versions are supported for 3-5 years.

Also download and install VirtualBox for your operating system. VirtualBox allows you to create and run virtual machines.

Creating the Virtual Machine

Open VirtualBox and click "New" to create a new virtual machine. Give it a name like "Ubuntu Test", select Linux and Ubuntu 64-bit.

Allocate enough RAM, ideally 2-3 GB if you have it. Create a new virtual hard disk of at least 10 GB.

Installing Ubuntu

In VirtualBox, select the Ubuntu virtual machine and click Start. When prompted, select the Ubuntu ISO file you downloaded. This will boot the Ubuntu installer.

Follow the prompts to install Ubuntu. Choose "Normal Install" unless you have bandwidth constraints. Partition the virtual hard drive carefully. Create a swap partition the same size as the RAM and an "ext4" root partition with the remaining space.

The install takes 30-40 minutes. You will set a username and password which you'll need to log in after install.

Finishing Up

Once installed, Ubuntu will reboot. Log in with the username and password you set.

You now have Ubuntu running in a virtual machine inside your existing OS! You can use the terminal and install additional software like Python for programming tasks.

A Beginner's Guide to Git and GitHub

What is Git and why use it?

Git is a version control system that allows you to track changes to your code over time. It allows you to experiment with new features and easily revert back if something goes wrong. Some key benefits of using Git:

- Track code changes over time with commits
- Easily collaborate with others by sharing code on GitHub
- Experiment freely knowing you can revert back to a working version
- Work on multiple features simultaneously using branches
- Avoid needing to make full copies of code each time you want a snapshot

Without Git, you would need to make full copies of your codebase each time you wanted a snapshot. This takes up a massive amount of storage space. Git uses diffs to only store the changes between commits, saving storage space.

Basic Git Commands

Here are some basic Git commands to get started:

- `git init` - Initializes a Git repository in the current folder
- `git status` - Shows the status of files in the repo, untracked or modified
- `git add` - Adds files to the staging area to include in the next commit
- `git commit` - Commits staged files with a message describing the changes
- `git log` - Shows a history of all commits in the repository

Collaborating with GitHub

GitHub provides remote hosting for Git repositories. This allows you to share your code with others and collaborate. Here's how to connect your local repo to a GitHub remote:

- Create a repository on GitHub
- Run `git remote add origin <url>` to link local repo to remote
- `git push origin master` pushes commits to the remote
- `git pull` fetches updates from the remote repo

GitHub provides a great web UI for viewing commit history, diffs, collaborating with others, and more.

Summary

- Git is a powerful version control system that allows tracking code history, branching, collaboration and more
- Basic commands: init, add, commit, status, log
- GitHub provides hosting for remote repositories to enable collaboration
- Following a basic Git workflow of adding/committing changes helps manage code

Tutorial 2

Week 2 Reddit Tutorial: Data Collection Overview

Introduction to Reddit

Reddit is a social media platform where users interact through comments and upvotes on posts. It has a community feel with subgroups called subreddits focused on specific topics. For example, r/olympics is about the Olympics. Users can search for keywords like "India" to find relevant subreddits, posts, communities and users.

Posts have titles, text bodies, images or videos. They accrue upvotes and comments. Comments allow layered conversations as users can reply to other comments.

Subreddits have rules, moderators, flairs (like hashtags), and related communities.

Collecting Reddit Data with PRAW

To collect Reddit data, first create a Reddit account and app. The app will provide authentication credentials like a client ID and secret.

Use the Python library PRAW to connect to Reddit with the credentials. The **`subreddit`** method specifies a particular subreddit to pull data from. For example, **`subreddit("india")`** collects posts from r/india.

PRAW returns post data like title, score, ID, URL, number of comments, created time, and body text. Store this in a Pandas dataframe for analysis.

Save collected data as CSV files to access later. The PRAW documentation explains available data fields like comments, usernames, post metadata etc.

Key Takeaways

- Reddit has subreddits, posts, comments, upvotes for user interaction
- Use PRAW and authentication credentials to collect Reddit data
- Pull post data like titles, scores, text, metadata into Python
- Save and analyze Reddit data frames with Pandas
- PRAW documentation lists available data fields to extract

Tutorial 3

Collecting and Storing Twitter Data in MySQL Database

Introduction

This tutorial covers how to collect Twitter data from the Twitter API and store it in a MySQL database. MySQL is an open-source relational database management system that uses SQL.

Installing MySQL

The first step is installing MySQL on your system using the `sudo apt-get install mysql-server` command. You will be prompted to set a root password during installation.

Then MySQL is configured using the `sudo mysql_secure_installation` command. This secures the installation by removing anonymous user accounts and remote root login.

Connecting Python to MySQL

To connect Python to MySQL, the `mysql-db` module needs to be installed using `sudo apt-get install python-mysql-db`.

This completes the installation and integration of MySQL with the Python environment.

Creating a MySQL Table to Store Tweets

A database called `osn_data` is created to store the tweets. Within this database, a table called `tweets` is created with `tweet_id` as the primary key and a `text` field to store the tweet content.

Streaming Tweets from Twitter API into MySQL

The Python script that collects real-time tweets from the Twitter API is modified to insert the tweets into the MySQL database.

The tweets are inserted using an `INSERT IGNORE` query with the `tweet_id` and `tweet_text` values. This adds new entries ignoring the duplicate primary keys.

Querying Stored Tweets from MySQL

After running the modified script for some time, the MySQL table contains multiple rows of tweets.

The `SELECT * FROM tweets` query is used to retrieve the stored tweets. Other queries like `SELECT COUNT(*) FROM tweets` can get the number of rows in the table.

This allows effective storage and querying of streaming Twitter data in MySQL database. The queries can be integrated into the Python script for better data collection and analysis.

Tutorial 4

Understanding Social Media Data as Network Graphs

Introduction

Social media platforms like Facebook and Twitter generate large amounts of data that can be analyzed as network graphs. Network graphs represent entities like users, pages, or groups as nodes, and relationships between them as edges. In this post, we will learn how to collect and visualize Twitter data as a network graph using tools like Twecoll and Gephi.

Representing Twitter Data as a Network Graph

There are different ways to represent a node-edge graph, including adjacency matrices, GraphML format, and CSV files.

An adjacency matrix is a 2D square matrix with dimensions equal to the number of nodes in the graph. A 1 in cell (i,j) indicates an edge exists between node i and node j. This is simple to construct using arrays in a programming language, but can take up a lot of space for sparse graphs.

GraphML is an XML-based format containing node and edge elements in sequence. Each node element must have a unique ID attribute and each edge element has source and target attributes identifying the endpoint nodes of an edge.

To collect Twitter data, we can use the Twecoll command line tool. It gathers information about your Twitter followers and "friends of friends." The **`python twecoll`** commandline prompts you to authorize a Twitter application and enter credentials. Then it retrieves follower IDs. **`python twecoll edgelist`** adds edges between you, your followers, and their followers.

Visualizing the Twitter Graph with Gephi

Gephi is an open-source tool for graph visualization. After installing it, we can open the GraphML file produced by Twecoll. Gephi displays summary statistics about the graph and can generate various network metrics like degree distribution, shortest paths, and modularity to find communities.

The graph layout can be customized by changing color, size, and shape of nodes and edges. We set the node color based on modularity class to see community structure and size by degree to highlight key nodes. The edge curvature can also be adjusted to improve readability. The final graph can be exported as an image or PDF file.

Key Graph Metrics

Some key metrics used in social network analysis:

- Degree - The number of edges incident to a node. Related measures are in-degree and out-degree.
- Centrality - Identifies important or central nodes. Common centrality measures are degree, closeness (mean shortest path length), and betweenness centrality.
- Modularity - Quantifies the strength of communities, based on the fraction of edges inside a given group.
- Data Laboratory - In Gephi, view the nodes, edges, and their attributes as originally stored in the GraphML file.

Conclusion

This summarizes the key steps covered in the video to collect Twitter data, represent it as a network graph, analyze graph metrics using Gephi, and customize the graph visualization.

Tutorial 5

Analyzing Text with NLTK

Overview

This video explains how to use the Natural Language Toolkit (NLTK) library in Python to analyze text data collected from Twitter. The goal is to clean and process the text to get insights into what the tweets are about.

Collecting and Preprocessing Tweets

The video starts by reading in 152 tweets collected about the Serum Institute. To analyze the tweets, the first step is to break them into individual words using NLTK's word tokenizer. This converts each tweet into a list of word tokens.

All the tokens from all the tweets are added to one large list. This list is passed to the Counter class from the Python collections module to get a count of each unique word.

Cleaning the Text

The most common words include some useful terms like "Serum" and "India" but also lots of noise like punctuation. To clean this up:

- All text is lowercased so "Serum" and "serum" map to the same word
- Punctuation is removed using Python's string translate() method
- Stopwords like "the", "of", "at" are removed using NLTK's stopwords corpus
- Tokens less than 2 characters are removed to delete residuals like "rt"

After this cleanup, the most common words provide a much clearer signal on what the tweets are about. Words like "fire", "vaccine", "Pune", "lives lost" indicate a fire at the Serum Institute facility in Pune that resulted in loss of life.

Key Takeaways

- NLTK provides useful text processing capabilities like tokenization and stopwords
- Cleaning the text by lowercasing, removing punctuation/stopwords, etc greatly improves analysis
- Even simple cleaning can reveal insights and topics within noisy text data like tweets

Tutorial 6

Using Gephi for Network Visualization

Introduction to Gephi

Gephi is an open source network analysis and visualization software used for research projects in education, journalism, digital humanities etc. It can import social network data from Facebook, Twitter etc. and generate graphs and clusters.

Loading Data into Gephi

- Gephi can read many file formats like gml, graphml, pajek net, uci net, dl files.
- To import from CSV, two files are needed - one with node list and one with edge list.
- Node CSV should have a column with unique node id.
- Edge list CSV should have source and target columns with node ids for each edge. Can also include a column indicating edge type - directed or undirected.

Overview Tab

The Overview tab provides options to customize the visualization:

- Appearance tab: Change node/edge color, size based on classical and continuous attributes.
- Layout tab: Choose from different layout algorithms like Fruchterman Reingold and customize them.
- Filters tab: Filter nodes and edges based on attributes like degree, followers etc.
- Statistics tab: Compute network metrics like average degree, diameter, modularity etc.

Preview Tab

The Preview tab shows updated visualization options for the generated graph. New layouts and filters can be tested without affecting existing workspace.

Data Laboratory

The Data Laboratory shows node and edge data as tables. Columns can be edited directly. Specific nodes/edges can be selected for analysis.

Customizing Visualization

Many customizations are possible:

- Change node color, size based on attributes like followers, degree etc.
- Modify edge color, thickness based on weight.
- Adjust labels, fonts, background color.
- Different layout algorithms like Fruchterman Reingold.
- Filter nodes by criteria like followers between a range.
- Create subsets as new workspaces using filters.

Key Features

- Import wide variety of network file formats.
- Interactive visualization with multiple customization options.
- Filter and analyze subsections of networks.

- Compute various network metrics like density, diameter, modularity etc.
- Export graphs as PDF, PNG, SVG.

Tutorial 7

Introduction to Data Visualization with Python and Highcharts

The video introduces how to create interactive data visualizations using Python scripts and Highcharts, a JavaScript charting library. It covers the following key concepts:

Creating Charts with Python Scripts

- Import the `highcharts` wrapper in Python to simplify generating visualizations
- Define a `container` to render the chart
- Set `chart options` like `chart type`, `title`, `xaxis`, `yaxis` etc to configure the chart
- Create `data arrays` with data points
- Add `data sets` along with `chart options` to the container
- Save the chart as HTML file that can be viewed in a browser

Four Types of Charts

The video demonstrates creating four types of charts from sample data:

Bar Chart

- Uses vertical/horizontal bars to represent grouped data
- Bar length represents the value proportionally

Line Chart

- Connects data points directly using straight lines
- Used to visualize data changing over time

Scatter Plot

- Pulls data points into individual dots, not connected
- Both axes represent some values like followers and friends

Bubble Chart

- Like scatter plot but bubble size represents additional quantity
- Can show 3 values: x-axis time, y-axis activity, z-axis volume

Using Highcharts Cloud

The Highcharts Cloud is a platform to instantly visualize data by just pasting it. It auto-generates charts. Benefits:

- Don't need to write any code
- Customizable charts
- Interactive features like filtering, tooltips
- Can save and download charts

Key Takeaways

- Python scripts using Highcharts wrapper allow programatically generating visualizations
- Highcharts Cloud provides easy interactive visualization without coding
- Many chart types available to visualize data based on purpose
- Visualizations make it easy to understand key insights from data