*A Progress Report*

*on*

**Human Behaviour Analysis Using Big Data Analytics**

*carried out as part of the course CSE CS3270 Submitted by*

***NAME:***

***REGESTRATION no. :***

***VI-CSE***

*in partial fulfilment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

In

**Computer Science and Engineering**

Text

Description automatically generated

**Department of Computer Science and Engineering,**

**School of Computer Science and Engineering,**

**Manipal University Jaipur,**

***JAN-MAY 2023***

**ACKNOWLEDGEMENT**

This project would not have completed without the help, support, comments, advice, cooperation and coordination of various people. However, it is impossible to thank everyone individually; I am hereby making a humble effort to thank some of them.

I acknowledge and express my deepest sense of gratitude of my internal supervisor Dr Chattar Singh Lambafor his/her constant support, guidance, and continuous engagement. I highly appreciate his technical comments, suggestions, and criticism during the progress of this project “ human behaviour analytics using big data analytics”

I owe my profound gratitude to **Prof. Neha Chaudhary**, Head, Department of CSE, for his valuable guidance and facilitating me during my work. I am also very grateful to all the faculty members and staff for their precious support and cooperation during the development of this project.

Finally, I extend my heartfelt appreciation to my classmates for their help and encouragement.

**Registration No. :**

**Student Name :**

Text

Description automatically generated

**Department of Computer Science and Engineering**

**School of Computer Science and Engineering**

Date: 18 August 2023

**CERTIFICATE**

This is to certify that the project entitled Human Behaviour Analytics using Big Data Analytics is a bonafide work carried out as ***Minor Project (Course Code: CS3270)***  in partial fulfilment for the award of the degree of Bachelor of Technology in Computer Science and Engineering, under my guidance by Abhishek Sharma bearing registration number 209301638 during the academic semester *VI of year 2022-23.*

**Place:** Manipal University Jaipur, Jaipur

**Signature of the project guide:**

**Name of the project guide: Dr Chattar Singh Lamba**

**Contents**

Page No.

Cover page

Certificate

Abstract

Table of Contents (with page nos)

1. Introduction
   1. Objective of the project
   2. Brief description of the project
   3. Technology used.
      1. H / W Requirement
      2. Software Requirement
   4. Organization Profile (if applicable)
2. Design Description
   1. Flow Chart
   2. Data Flow Diagrams(DFDs)
   3. Entity Relatoinship Diagram(E-R Diagram)
   4. Use-Case Diagram
   5. Class Diagram
   6. Sequence Diagram
   7. Activity Diagram
3. Project Description
   1. Data Base
   2. Table Description
   3. File/Database Design
4. Input/Output Form Design
5. Testing & Tools used (if applicable)
6. Implementation & Maintenance (if applicable)
7. Future scope
8. Conclusion
9. Bibliography

**INTRODUCTION**

1. **OBJECTIVE(S) OF PROJECT**

The main objectives of our project are as follows:

* Use NLP techniques to extract valuable insights from vast amounts of Reddit user data. Specifically, we aim to identify the sentiment, topics, and trends that are prevalent in the Reddit community. The analysis will be conducted on a large corpus of Reddit comments and posts to gain a comprehensive understanding of user behavior and preferences.
* Identify whether a given user is Bot or not using Decision Tree classifier to differentiate actual opinions to those of little to no face value.
* Build intelligent agents that can learn from the data and make informed decisions. These agents will be able to perform a variety of tasks such as recommending content, identifying potential influencers, and predicting user behavior. We have an arbitrary formula which will be used in generating
* Provide valuable insights to businesses, advertisers, and marketers to inform their strategies and decision-making especially to targeted audiences.

1. **BRIEF DESCRIPTION OF PROJECT**

* What is Behavioural Analytics?

Behavior analytics is an emerging field that is becoming increasingly popular in the programming industry. Behavioral analytics is the process of recording user activity, grouping the data into behavioral cohorts, and then analyzing relevant metrics, such as user engagement with the product.

The insights you get from behavioral data help you understand how user engagement with your product is related to increasing engagement, retention, lifetime value, conversion rates, and ultimately revenue. Behavioral analytics tools generally measure user actions or behavior by firing custom events or clicks on the front end. It also allows you to identify the issues that your users face with your product and helps you find scopes of improving the user experience.. There are various types of behavior analytics, including web analytics, user behavior analytics, application analytics, and mobile analytics, among others.

One of the main advantages of behavior analytics is that it provides organizations with a way to identify patterns and trends in user behavior, which can be used to improve products and services. For example, by analyzing user behavior on a website, businesses can gain insights into what pages are most popular, which features are most frequently used, and what factors contribute to user engagement. This information can then be used to optimize the website and create a more engaging user experience.

Behavior analytics is currently being used in a variety of applications, including e-commerce, healthcare, security, finance, and gaming. In e-commerce, for example, companies are using behavior analytics to track customer behavior and optimize their shopping experience. In healthcare, behavior analytics is being used to improve patient outcomes by analyzing patient behavior and identifying areas where interventions can be made to improve health outcomes.

Despite its many advantages, behavior analytics also presents a number of challenges. One of the main challenges is ensuring the privacy and security of user data. This requires organizations to implement strong data security measures and adhere to strict data privacy regulations. Additionally, behavior analytics requires specialized skills and expertise, which can be a challenge for organizations without the necessary resources or talent.

* What is Natural Language Processing?

Natural language processing (NLP) refers to the branch of computer science concerned with giving computers the ability to understand text and spoken words in much the same way human beings can. NLP combines computational linguistics—rule-based modeling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to ‘understand’ its full meaning, complete with the speaker or writer’s intent and sentiment. Examples of NLP that we use on a regular basis are voice-operated GPS systems, digital assistants, speech-to-text dictation software or customer service chatbots.

* What is our project and how does it tie these technologies together?

Our project is unique in the fact that it will be combining all the aforementioned technologies together in a structured manner to get results. We will be using data on Reddit users via the Reddit application programming interface (API) namely: Top Posts on subreddit (per day/week/month), Popular Posts on subreddit (per day/week/month), Rising Posts on subreddit (per day/week/month), Post Details, Top Comments and Posts by Username and Average Karma (Likes – Dislikes).

We then use a model trained specifically on Reddit comments to filter out the bots. The way we do this is we take a look at the several JSON fields attached to all Reddit comments, namely:

no\_follow  
link\_id  
gilded  
author  
author\_verified  
author\_comment\_karma  
author\_link\_karma  
num\_comments  
created\_utc  
score  
over\_18  
body  
is\_submitter  
controversiality  
ups  
is\_bot  
is\_troll  
recent\_comments

along with added sentiment polarity fields (negative/positive) for more robust calculation. We plug these values in a decision tree classifier (we are using a decision tree classifier because it is easier to understand and implement at the cost of some accuracy) and we assign weights to the fields to find out whether or not a username is a bot/troll or an actual regular user. For example if a user has a brand new account (< 2 weeks) and are submitting posts at a rapid rate with no comments and little karma then most likely, they are a bot. If a username is a bot then we dump all the data and if not then we proceed further.

Our next step involves the behavior analysis part of our project. This comprises of four main levels/components.

(i)The upvotes/downvotes (karma) of posts and comments in a subreddit

(ii)The SR Reach Score; which is an arbitrary score comprising of no of followers of a subreddit, average Gunning Fog Index and Corpus Statistics; to figure out the accessibility and userbase proficiency of the subreddit.

(iii)Sentiment Analysis of Posts

1. **TECHNOLOGY USED**
2. **Hardware Requirements**

**Development Hardware**

1. Processor: i5-10300H (8 Cores) @ 2.5 GHz

2. Hard Drive: 512 GB SSD

3. Memory (RAM): 12 GB

4. Graphics: NVIDIA 1650 ti

4. Ethernet Connection (LAN) and WiFi Connection Card

5. Sound Card w/ speakers

**Deployment Hardware**

**Minimum**

1. Processor: i3-7020U (2 Cores) @ 2 GHz or equivalent

2. Hard Drive: 256 GB or More (SSD Preferred)

3. Memory (RAM): 4 GB

4. Graphics: NVIDIA GTX 1050 or equivalent

5. Ethernet Connection (LAN) and WiFi Connection Card

6. Sound Card w/ speakers

**Recommended**

1. Processor: i5-8260U (4 Cores) @ 3.9 GHz or equivalent

2. Hard Drive: 512 GB or More (SSD Preferred)

3. Memory (RAM): 6 GB

4. Graphics: GTX 1050 or equivalent

4. Ethernet Connection (LAN) and WiFi Connection Card

5. Sound Card w/ speakers

**b. Software Requirements**

**Operating Systems (Recommended)**

**Windows:** 7 or newer

**MAC:** OS X v10.7 or higher

**Linux:** Ubuntu

Web Browser: Google Chrome, Mozilla Firefox, or Safari

Programming Languages: Python, HTML, CSS, and JavaScript

Python Libraries: Pandas, NumPy, Matplotlib, NLTK, Scikit-learn, Flask, Gunicorn

Big Data: Hadoop Mapreduce, Apache Spark

Web Framework: Flask

Database: MongoDB

Version Control: Git

Editor: Visual Studio Code

**c. Functional Requirements**

Functional Requirements The project system will have the following functional requirements:

1. Input Interface: The system should have an input interface that allows users to input Reddit post URLs, select the subreddit name, date range, and specify the behavior or emotion they want to analyze.

2. Data Collection: The system should be able to collect data from Reddit comments based on the input provided by the user. The system should be able to collect large amounts of data from multiple Reddit posts simultaneously.

3. Data Preprocessing: The system should preprocess the collected data by removing stop words, stemming, and tokenization.

4. Sentiment Analysis: The system should perform sentiment analysis on the preprocessed data to identify positive, negative, or neutral comments.

5. Topic Modeling: The system should perform topic modeling on the preprocessed data to identify the most relevant topics discussed in the comments.

6. Network Analysis: The system should perform network analysis on the preprocessed data to identify the most influential users and their connections.

7. Visualization: The system should provide users with a visual representation of the analysis results using graphs, charts, or word clouds.

8. Download Raw Data: The system should provide users with the option to download the raw data used for analysis.

Non-Functional Requirements:- The Project system will have the following non-functional requirements:

1. Performance: The system should be able to handle large amounts of data and perform analysis in a reasonable amount of time.

2. Accuracy: The system should provide accurate analysis results to ensure that users can make informed decisions based on the data.

3. Security: The system should ensure that user data is secure and protected from unauthorized access.

4. Usability: The system should be user-friendly, easy to navigate, and intuitive to use.

**DESIGN DESCRIPTION**

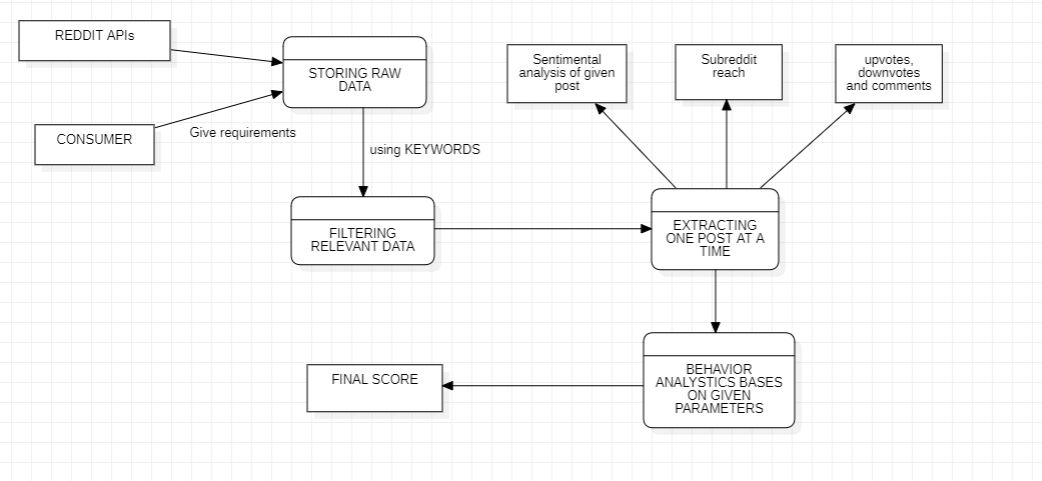
* 1. **FLOWCHART**

**Diagram

Description automatically generated**

* 1. **DATA FLOW DIAGRAM (DFDs)**

**Level 0 DFD**

****

**Level 1 DFD**

**Diagram

Description automatically generated**

* 1. **Diagram

     Description automatically generatedENTITY-RELATIONSHIP DIAGRAM (E-R DIAGRAM)**
  2. **USE CASE DIAGRAM**

Diagram, schematic

Description automatically generated

* 1. **CLASS DIAGRAM**

Diagram

Description automatically generated

* 1. **SEQUENCE DIAGRAM**

A picture containing diagram

Description automatically generated

* 1. **ACTIVITY DIAGRAM**

Diagram

Description automatically generated

Diagram

Description automatically generated

**INPUT/OUTPUT FORM**

Input Form:

Reddit Post URL: [Insert a text box where users can enter the URL of the Reddit post they want to analyze]

Subreddit Name: [Insert a text box where users can enter the name of the subreddit where the post is located]

Date Range: [Insert two date pickers where users can select the start and end dates for the comments they want to analyze]

Keyword(s): [Insert a text box where users can enter one or more keywords related to the behavior they want to analyze]

Analysis Type: [Insert a dropdown menu where users can select the type of behavior analysis they want to perform (e.g., sentiment analysis, topic modeling, network analysis)]

Output Form:

Summary Report: [Display a summary report of the analysis results, including the number of comments analyzed, the overall sentiment or topics discussed, and any insights related to the behavior being analyzed]

Visualization: [Display a visualization (e.g., a graph, chart, or word cloud) of the analysis results to help users quickly understand the key findings]

Raw Data: [Provide a download link or button that allows users to download the raw data used for the analysis, in case they want to perform their own analysis or further investigate the results]

Share Results: [Include a button or link that allows users to share the analysis results on social media or with others, if they want to spread awareness or discuss the findings with a wider audience]

**DESIGNING TEST SUITES AND ESTIMATION OF TEST COVERAGE METRICS**

Testing is the process of evaluating a system or its component's with the intent to find that whether it satisfies the specified requirements or not .This activity results in the actual, expected and difference between their results i.e testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements.

**1.Testing Strategies**

In order to make sure that system does not have any errors, the different levels of testing strategies that are applied at different phases of software development are

1.1 Unit Testing

The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

1.2 Integration Testing

The testing of combined parts of an application to determine if they function correctly together is Integration testing. This testing can be done by using two different methods

1.2.1 Top Down Integration testing

In Top-Down integration testing, the highest-level modules are tested first and then progressively lower-level modules are tested.

1.2.2 Bottom-up Integration testing

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules.In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing.

1.3 System Testing

This is the next level in the testing and tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets Quality Standards.

1.4 Acceptance Testing

The main purpose of this Testing is to find whether application meets the intended specifications and satisfies the client's requirements. We will follow two different methods in this testing.

1.4.1 Alpha Testing

This test is the first stage of testing and will be performed amongst the teams Unit testing, integration testing and system testing when combined are known as alpha testing. During this phase, the following will be tested in the application:

Spelling Mistakes.

Broken Links.

The Application will be tested on machines with the lowest specification to test loading times and any latency problems.

1.4.2 Beta Testing

In beta testing, a sample of the intended audience tests the application and send their feedback to the project team. Getting the feedback, the project team can fix the problems before releasing the software to the actual users.

**2. Testing Methods**

2.1 White Box Testing

White box testing is the detailed investigation of internal logic and structure of the Code. To perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code .The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

An example of White Box Testing being performed on our project:

1. Test case for the implementation of sentiment analysis algorithm:

* Input: A comment with positive sentiment.
* Input: A comment with negative sentiment.
* Input: A comment with neutral sentiment.
* Expected Output: The sentiment analysis algorithm correctly assigns positive, negative, and neutral sentiment to the comments.

1. Test case for the implementation of data processing algorithms:

* Input: A large dataset containing comments from multiple subreddits.
* Expected Output: The data processing algorithms are able to efficiently extract relevant data points such as sentiment, karma score, and number of comments.

1. Test case for the implementation of machine learning models:

* Input: A dataset containing labeled comments with sentiment scores.
* Expected Output: The machine learning models are able to accurately predict the sentiment of new comments based on the features extracted from the data.

1. Test case for the implementation of the recommendation system:

* Input: A set of user preferences and product features.
* Expected Output: The recommendation system is able to accurately suggest relevant subreddits to advertise the product based on the user preferences and product features.

1. Test case for the implementation of the user interface:

* Input: User interaction with the system through the interface.
* Expected Output: The interface is intuitive and user-friendly, with clear instructions and feedback provided to the user.

2.2 Black Box Testing

The technique of testing without having any knowledge of the interior workings of the application is Black Box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

An example of black box testing via Boundary Value Analysis is as follows:

1. Test case for the number of posts in the subreddit:

- Input: The subreddit contains 4999 posts.

- Input: The subreddit contains 5000 posts.

- Input: The subreddit contains 5001 posts.

- Input: The subreddit contains 5500 posts.

- Input: The subreddit contains 6000 posts.

2. Test case for the length of comments in the subreddit:

- Input: A comment has 1 character.

- Input: A comment has 10 characters.

- Input: A comment has 100 characters.

- Input: A comment has 1000 characters.

- Input: A comment has 5000 characters.

3. Test case for the sentiment score of posts in the subreddit:

- Input: A post has a sentiment score of -1.

- Input: A post has a sentiment score of 0.

- Input: A post has a sentiment score of 1.

- Input: A post has a sentiment score of 2.

- Input: A post has a sentiment score of 3.

4. Test case for the karma of comments in the subreddit:

- Input: A comment has a karma score of -1.

- Input: A comment has a karma score of 0.

- Input: A comment has a karma score of 1.

- Input: A comment has a karma score of 100.

- Input: A comment has a karma score of 1000.

These test cases cover a range of input values for different variables related to the r/pics subreddit, and can help identify potential issues with the sentiment analysis system.

1. Test case for the karma score of individual members:

- Input: A member has a karma score of -100.

- Input: A member has a karma score of 0.

- Input: A member has a karma score of 1.

- Input: A member has a karma score of 1000.

- Input: A member has a karma score of 10000.

2. Test case for the sentiment score of individual members' comments on a certain product:

- Input: A member has a sentiment score of -1 for a certain product.

- Input: A member has a sentiment score of 0 for a certain product.

- Input: A member has a sentiment score of 1 for a certain product.

- Input: A member has a sentiment score of 2 for a certain product.

- Input: A member has a sentiment score of 3 for a certain product.

3. Test case for the number of comments made by individual members on a certain product:

- Input: A member has made 1 comment on a certain product.

- Input: A member has made 5 comments on a certain product.

- Input: A member has made 10 comments on a certain product.

- Input: A member has made 50 comments on a certain product.

- Input: A member has made 100 comments on a certain product.

4. Test case for the sentiment score of comments on a certain product in the subreddit:

- Input: A comment has a sentiment score of -3 for a certain product.

- Input: A comment has a sentiment score of -1 for a certain product.

- Input: A comment has a sentiment score of 0 for a certain product.

- Input: A comment has a sentiment score of 1 for a certain product.

- Input: A comment has a sentiment score of 3 for a certain product.

By testing with different values for these input variables, we can ensure that our sentiment analysis system is able to handle a wide range of inputs and produce accurate results. This can help identify potential issues with the system, such as sensitivity to certain types of inputs or limitations in the range of input values that can be processed.

**3. Validation**

All the levels in the testing (unit,integration,system) and methods (black box,white box)are implemented on our application successfully and the results obtained as expected.

1. **Limitations**

There are a lot of complex algorithms at play here not to mention several independent third party services and applications being used so a single request may take some time.

**5. Test Results**

The testing is done among the team members and by the end users. It satisfies the specified requirements and finally we obtained the results as expected.

**Test Suites**

1. Data processing test suite:

* Test case 1: Verify that the system can handle large datasets containing comments from multiple subreddits.
* Test case 2: Verify that the system is able to extract relevant data points such as sentiment, karma score, and number of comments from the dataset.
* Test case 3: Verify that the system can handle missing or incomplete data points and still produce accurate results.
* Test case 4: Verify that the system can handle noisy or irrelevant data points and still produce accurate results.

1. Sentiment analysis test suite:

* Test case 1: Verify that the sentiment analysis algorithm can correctly assign positive, negative, and neutral sentiment to comments.
* Test case 2: Verify that the sentiment analysis algorithm can handle sarcasm, irony, and other forms of figurative language commonly used in online discussions.
* Test case 3: Verify that the sentiment analysis algorithm can handle multilingual comments and produce accurate results across different languages.
* Test case 4: Verify that the sentiment analysis algorithm can handle slang, abbreviations, and other informal language commonly used in online discussions.

1. Machine learning test suite:

* Test case 1: Verify that the machine learning models can accurately predict the sentiment of new comments based on the features extracted from the data.
* Test case 2: Verify that the machine learning models can handle imbalanced datasets and produce accurate results for rare classes.
* Test case 3: Verify that the machine learning models can handle new and unseen types of comments and produce accurate results.
* Test case 4: Verify that the machine learning models can handle noisy or irrelevant data points and still produce accurate results.

1. Recommendation system test suite:

* Test case 1: Verify that the recommendation system can accurately suggest relevant subreddits to advertise the product based on the user preferences and product features.
* Test case 2: Verify that the recommendation system can handle noisy or irrelevant data points and still produce accurate results.
* Test case 3: Verify that the recommendation system can handle missing or incomplete data points and still produce accurate results.
* Test case 4: Verify that the recommendation system can handle large datasets and produce results in a timely and efficient manner.

1. User interface test suite:

* Test case 1: Verify that the user interface is intuitive and user-friendly, with clear instructions and feedback provided to the user.
* Test case 2: Verify that the user interface is accessible and compatible across different platforms and devices.
* Test case 3: Verify that the user interface can handle different types of user inputs and produce accurate results.
* Test case 4: Verify that the user interface is visually appealing and consistent with the branding and design guidelines of the product.

**TESTING TOOLS LIKE SELENIUM AND LOAD RUNNER**

Selenium is a popular open-source testing tool used for automating web browsers. It enables software testers and developers to write and run automated tests for web applications in various programming languages such as Java, Python, C#, Ruby, and more.

Selenium provides a range of features and functionalities, including:

Record and Playback: Selenium allows users to record their actions while manually testing a web application and then play them back to automate the same tasks.

Cross-Browser Compatibility: Selenium supports testing of web applications across different browsers such as Chrome, Firefox, Safari, Edge, and Internet Explorer.

Test Script Generation: Selenium provides a mechanism to generate test scripts in various programming languages, which can be used to automate functional testing, regression testing, and other types of testing.

Integration with Testing Frameworks: Selenium can be integrated with various testing frameworks such as TestNG, JUnit, NUnit, and more to help manage and organize test cases.

Parallel Test Execution: Selenium supports parallel test execution, allowing testers to run multiple test cases simultaneously, reducing the overall testing time

LoadRunner is a performance testing tool developed by Micro Focus that allows users to test the performance and scalability of applications under heavy load. LoadRunner simulates real-world user traffic by creating virtual users that mimic the actions of actual users.

LoadRunner works by capturing and analyzing network traffic between the client and server components of the application. It then generates virtual users that replicate the actions of real users, such as clicking on links, filling out forms, and submitting data. These virtual users can be customized to mimic various user profiles and load scenarios.

LoadRunner provides a range of performance testing features, including:

Load Testing: LoadRunner allows users to simulate thousands of virtual users and measure the application's performance under heavy load.

Stress Testing: LoadRunner can help identify the breaking point of an application by pushing it to its limits and measuring its performance under high levels of stress.

Endurance Testing: LoadRunner helps users to identify how the application performs over a sustained period of time, such as a 24-hour period.

Capacity Planning: LoadRunner can help users to determine the maximum capacity of an application, based on various load scenarios and user profiles.

Performance Analysis: LoadRunner provides various performance metrics such as response times, throughput, and resource utilization, which help identify performance bottlenecks and optimize the application.

**LIBRARIES / MODULES USED**

* **Pandas:** pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.
* **NumPy:** NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
* **Matplotlib:** Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.
* **NLTK:** The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.
* **Scikit-learn:** scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language.[3] It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is a NumFOCUS fiscally sponsored project.
* **Json:** Python JSON JavaScript Object Notation is a format for structuring data. It is mainly used for storing and transferring data between the browser and the server. Python too supports JSON with a built-in package called JSON. This package provides all the necessary tools for working with JSON Objects including parsing, serializing, deserializing, and many more.
* **PRAW:** PRAW (Python Reddit API Wrapper) is a Python module that provides a simple access to Reddit’s API. PRAW is easy to use and follows all of Reddit’s API rules.
* **Diff lib:** Difflib is a built-in module in the Python programming language consisting of different simple functions and classes that allow users to compare data sets. The module offers the outputs of these sequence comparisons in a format that can be read by a human, using deltas to show the differences more efficiently.
* **Psycopg2:** Psycopg is the most popular PostgreSQL adapter used in Python. Its works on the principle of the whole implementation of Python DB API 2.0 along with the thread safety (the same connection is shared by multiple threads). It is designed to perform heavily multi-threaded applications that usually create and destroy lots of cursors and make a large number of simultaneous INSERTS or UPDATES. Psycopg features client-side and server-side cursors, asynchronous communication, and notification. Psycopg 2 is both Unicode and Python 3 friendly.
* **text blob:** TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.
* **Seaborn:** Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
* **Ipython:** Python is a command shell for interactive computing in multiple programming languages, originally developed for the Python programming language, that offers introspection, rich media, shell syntax, tab completion, and history.
* **Flask:** Flask is a web framework, it’s a Python module that lets you develop web applications easily. It’s has a small and easy-to-extend core: it’s a microframework that doesn’t include an ORM (Object Relational Manager) or such features. It does have many cool features like url routing, template engine. It is a WSGI web app framework.

**IMPLEMENTATION AND MAINTAINENCE**

1. **Diagram

   Description automatically generatedWorking of Bot detection**

Trolls and bots are widespread across social media, and they influence us in ways we are not always aware of. Trolls can be relatively harmless, just trying to entertain themselves at others’ expense, but they can also be political actors sowing mistrust or discord. While some bots offer helpful information, others can be used to manipulate vote counts and promote content that supports their agenda. We’ll show you how machine learning can help protect our communities from abuse.

we need to select features to build our model. Reddit provides dozens of JSON fields for each user and comment. Some don’t have meaningful values. For example, banned\_by was null in every case, probably because we lack moderator permissions. We picked the fields below because we thought they’d be valuable as predictors or to understand how well our model performs. We added the column recent\_comments with an array of the last 20 comments made by that user.

no\_follow  
link\_id  
gilded  
author  
author\_verified  
author\_comment\_karma  
author\_link\_karma  
num\_comments  
created\_utc  
score  
over\_18  
body  
is\_submitter  
controversiality  
ups  
is\_bot  
is\_troll  
recent\_comments

Our next step is to create a new machine learning model based on this list. We’ll use Python’s excellent [scikit learn](https://scikit-learn.org/) framework to build our model. We’ll store our training data into two data frames: one for the set of features to train in and the second with the desired class labels. We’ll then split our dataset into 70% training data and 30% test data.

Next, we’ll create a [decision tree classifier](https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html) to predict whether each comment is a bot, a troll, or a normal user. We’ll use a decision tree because the created rule is very easy to understand. The accuracy would probably be improved using a more robust algorithm like a random forest, but we’ll stick to a decision tree for the purposes of keeping our example simple.

FOLLOWING RESULTS ARE OBTAINED:

**Table

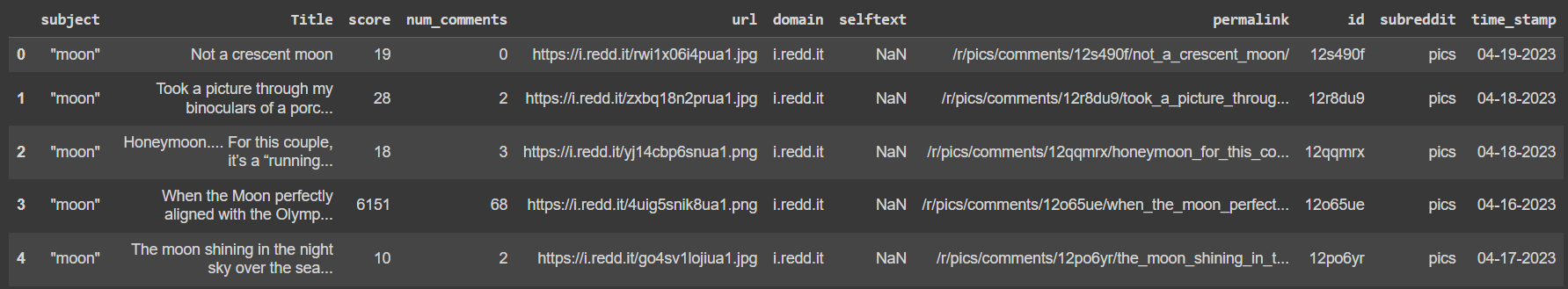
Description automatically generated**

**Text

Description automatically generated**

**Working of sentimental analysis:**

The goal is to find the Trends and opinions on an idea or product this is achieved by doing the following steps:

1. Pulling the data of Reddit using Reddit API and PRAW.
2. Storing data into a file with subject, title, score num\_comments, url, domain, selftext, permalink, id, subreddit, time\_stamp.
3. Now, we filter the data using the keywords and we can also choose from which sub-reddit we want to extract the data
4. We will use the vaderSentiment to analyze the sentiment of the post and classify them as positive, neutral and negative.
5. Finally, we count the number of positive negative and neutral posts from filtered data

Text

Description automatically generated

We can now calculate the Sub-Reddit reach score using the following formula:

**SR REACH SCORE = Gunning Fogg index \* (pos+neu-neg) \* (no of subscribers of subreddit \* 0.0001)**

This score helps us determine how appropriate the SubReddit will be to advertise our product or how the general public feels about the given idea. The information gathered will be our final analytic report the customer can see the whole data and can make his own inference if he wants to.

Graphical user interface, text, application, email

Description automatically generated

**FUTURE SCOPE**

The future scope of the project is massive as it utilizes top of the line technology. This project was done on Reddit userbase but can be integrated into several new platforms. We can further expand the scope of the analysis if needed by integrating trending topics and real-time analysis of events. We can also further improve the project by incorporating more advanced classification and regression techniques. We can also improve the project by adding in more data visualization in the form of graphs and charts. However, I have confidence that our project will be able to serve as the starting point of a new way to look at ABA.

**CONCLUSION**

The use of big data in the field of social media is essential. The organizations that use big data have a huge advantage over the one which is still practicing relational database techniques. These organizations better know the importance of big data than the one which has no big data implementation. This product is intended to show the behavioral analysis of Reddit users and some key results. In this way, many other features can be determined, and the company could know the details of its competitor and clients. If a company markets its product on Reddit, its product becomes more prominent than the base of views and likes.

**BIBLIOGRAPHY**

* Grover, V., Lindberg, A., Benbasat, I., and Lyytinen, K. (2020). The perils and promises of big data research in information systems. J. Assoc. Inf. Syst. 21:9
* Lydia, E. L., and Swarup, M. B. (2016). Analysis of big data through hadoop ecosystem components like flume, mapreduce, pig and hive. Int. J. Comput. Sci. Eng. 5, 21–29
* Mahalakshmi, R., and Suseela, S. (2015). Big-SoSA: social sentiment analysis and data visualization on big data. Int. J. Adv. Res. Comp. Commun. Eng. 4, 304–306.
* Paul, A., Ahmad, A., Rathore, M. M., and Jabbar, S. (2016). Smartbuddy: defining human behaviors using big data analytics in social internet of things. IEEE Wirel. Commun. 23, 68–74. doi: 10.1109/mwc.2016.7721744
* Rodrigues, A. P., Rao, A., and Chiplunkar, N. N. (2017). “Sentiment analysis of real time Twitter data using big data approach,” in Proceedings of the 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), (Piscataway, NJ: IEEE), 1–6
* . Iqbal, R., Doctor, F., More, B., Mahmud, S., and Yousuf, U. (2020). Big data analytics: computational intelligence techniques and application areas. Technol. Forecast. Soc. Change 153:119253. doi:10.1016/j.techfore.2018. 03.024
* Stieglitz, S., Mirbabaie, M., Ross, B., and Neuberger, C. (2018). Social media analytics–Challenges in topic discovery, data collection, and data preparation. Int. J. Inf. Manag. 39, 156–168. doi: 10.1016/j.ijinfomgt.2017.12.002