EUROPEAN UNIVERSITY OF LEFKE

FACULTY OF ENGINEERING

Graduation Project I

Sentiment Analysis and Text Classification using Mlpack C++

JOHN KELECHUWKU OBI

20140237

Sentiment analysis of textual data used to assess products in discussion forums and e-commerce settings is the main objective of this web-based platform project. It makes an effort to perform a thorough text classification. It will be used to help arrange text based on sentimental significance in order to quickly support the clients who are providing feedback. Finding the text's emotions, whether favourable or negative, is the goal.

Supervisor

Asst. Prof. Dr. Cem Kalyoncu

Publish Date

21/12/2023

Table Of Contents

1.Introduction

1.1 Problem definition

The aim is to address the challenge of analyzing the sentiment in text data. Sentiment analysis involves determining the emotional tone expressed in a piece of text, classifying it as positive or negative. This is crucial for understanding user and public opinions, customer feedback and social media sentiment. Where the problem lies is in accurately categorizing diverse textual inputs which may contain subtle nuances and context-dependent sentiments. The challenge is to develop a robust system that can with less issues discern sentiments effectively and provide valuable insights into the emotional tone of the text. Large text volumes are time consuming to analysis and very easily prone to bias.

The complexity of this problem arises from the inherent subjectivity and variability of human language. Different people may express the same sentiment in different ways and a single individual may express their sentiments differently at different times. [1]

The project's objective is to create an accurate text classification system with a user interface that can easily integrate with third party platforms.

The project follows four stages; data collection, cleaning and preprocessing where feature engineering techniques are applied; building and comparing machine learning models using ML methods supported by Mlpack.

The hosting phase involves serving the model, over HTTPS to allow access while the web server phase involves communication, with the model server and serving users.

Example-Problems:

- **User-unfriendliness;** From the standpoint of the end user, interacting directly with most machine learning models can be somewhat complicated; this project streamlines the interaction process. I'm referring to the hugging face model platform, where using a model might be really difficult to begin with.

- **Integration;** Most machine learning model are also quite hard to integrate into an already existing application and those that can may involve adding and going through twisted procedures just to have a real-time text classification.

- **Huge data**; Having a database of stored key words that includes negative terms was the conventional or outdated method of completing this assignment. This becomes more expensive if there is a lot of data in the database and you have to go through lengthy rows of texts word by word to analyse a text.

1.2 Goals

- **Real time usage:**
  The ability to perform or provide real time sentiment analysis and text classification. Crucial for social media apps and business apps where users feedback is valuable as quick as possible.

- **Cost reduction**;
  Just taking the traditional way into consideration, the act of storing bad words and iterating through the rows word for word to compare is too time and resource consuming and not cost efficient. This project goal is to minimize such cost.

- **User friendly and Integration friendly**;
  Utilize dynamic HTML content to create an expressive and intuitive frontend experience that is easy for developers and end users to utilize.

- **Accurate and automated sentiment analysis**
  To be able to develop a system that have high accuracy in sentiment analysis, ensuring reliable text classification of positive and negative sentiments.

- **Api endpoints**

  To create a robust backend to handle Api endpoints for data collection and model interaction and database updates. Enables seamless communication between the frontend and the ml models.

- **Database integration, data storage and retrieval**

  Integrate a database, in the system we use PostgreSQL to store and manage textual data and model predictions and additional information for historical tracking and retrieval.

- **Containerization and consistent development environment**

  To be able to employ docker for testing, development and deployment ensuring consistency and reproductivity across different environments.

2. Literature Survey

Eden Ai

It is a platform that offers unified Api from multiple Ai providers and provides an interface for businesses and developers who want to incorporate Ai into their systems. [2]

My project's dev-friendly model access is similar to Eden AI's. The distinction is that my project solely focuses on review sentiment solutions, which allows my system to meet the demands and specifications of sentiment analysis tasks.

For instantaneous analysis While it's unclear if Eden AI enables real-time analysis, my project offers real-time analysis.

In terms of user-friendliness, Eden AI currently only offers an API for developers or other non-end users, whereas my project uses an interactive web interface to serve both immediate users and end users.

Eden ai provides the opportunity to build custom ai powered chatbot, customize the model’s features, add custom data collection whilst mine only provide sentiment text classification. [2] Majority of the models used by Eden ai is mainly powered by python whilst mine uses C++.

Levity Ai

This platform very similar to mine and also focuses more on automated workflows using AI whilst mine do not. We share similarities in integration and usage of AI capabilities. One noticeable difference is Levity AI has a sign in policy while mine don’t.

With further in depth digging and research about levity AI and its tech stack tools, a difference of toolsets is also among what differentiate my project with theirs. They utilize a lot of open-source models and fine tune it whilst mine is built from scratch. [3]

Hugging face

Hugging face is an open-source project that builds and deploys pre-trained models. They enable the download of already pre-trained models mine does not. Has a platform where researchers, students and developers share their work. They provide wider range of tools whilst mine is more lightweight and focus on a tiny little sector of what hugging face serves. The models served by hugging face are very complex to use and also to integrate you will need a deeper understanding of NLP concepts and python programming and bash skills before you can integrate it into your system.

The goal of hugging face is also very different from mine and the previous compared platform. They are on a mission to democratize good machine learning, one commit at a time. [4]

3. Background Information

3.1 Required software

- **Typescript**
  A strongly typed programming language for JavaScript that helps set rules in order to reduce bugs and runtime errors that stems from JavaScript dynamic nature. [5]

- **C++**
  It is a well-known robust, powerful and efficient language. İt was picked to be able to handle the real time analysis of the system with low latency and also was picked due to the needs by my supervisor.

- **Mlpack Library**
  Due to the use of C++ programming language, arises the need of a machine learning library in C++ and Mlpack is a lightweight, fast, header-only library. Once you have developed a machine learning workflow, deployment is straightforward. Mlpack directly depends on only three libraries, Armadillo, Ensmallen and cereal. [6]
  It's quite mature and has support for use in production settings.

- **Uwebsockets**
  a C++ library that is lightweight and used for building real-time web servers with high performance and scalable solution. Its best features is been fast, easy to integrate into existing projects and memory efficient. [7] This is the recommended C++ server library to host the Mlpack model but this project will only rely on the HTTP part of the library instead on the WebSocket.

- **Nlohmann json library**
  Json library that supports manipulation of Json data in C++ because most of the data collected will be in Json file format therefore this library is needed to handle them.

- **ExpressJS:**
  A JavaScript library for building backend servers and Api routes and capable of running in almost all JavaScript runtimes be it bun, node, deno, etc.

- **PostgreSQL C++ library**
  To connect your C++ application with PostgreSQL database, you would need a C++ library. This library provides an API to interact with PostgreSQL.

- **EJS engine**
  this is a JavaScript templating engine, runs with any JavaScript runtime that is use to generate dynamic html content for the web.

- **PostgreSQL**
  A robust modern database for the system. This database is picked due to the cloud hosting platform's easy Postgres integration.

3.2 Other software

- **Adobe Illustrator**
  Adobe Illustrator is a vector graphics editor and design program developed and marketed by Adobe Inc. [8]
  Use to create mock-up of the web interface design and draw the use case diagrams and workflows of the project.

- **Visual Studio Code**
  A code editor that supports multiple languages and have remote ssh capabilities to build and develop inside dockerized containers in isolated environments.

- **Matplotlibcpp**
  A C++ header wrapper for a python graphing software library to help visualize the dataset and performance stats during model training. Matplotlibcpp wraps around

matplotlib which is a library for create interactive, static or animated visuals of datasets. [9]

- **Docker**
  A containerisation platform that is used to set up the development environments for building the projects, where I use visual studio code to remote access into an isolated docker container, package and deploy the software. This is one of the best tools I love to use. It helps you test and try different OS images and find the optimal image that just does what you want and no more.

- **Cmake**

  A building system that is used to build the various C/C++ libraries for both the development environments and production environments.

  It is the de-facto standard for building C++ code, it is comprehensive, powerful solution for managing the build process. [10]

- **Git**
  A version control system to track the project's progress and development's history. This also helps me roll back to previous working state of the project when errors appear.

4. Modules

4.1 Training Module

This module is used by the developer and is responsible for collecting and preprocessing the review data, training the text classification model using Mlpack library.

4.1.1 Data Collection and preprocessing

This submodule collects data for training from various source but for this project it's from a specific source. Cleans, breaks the test down into individual tokens/text by a tokenizer type and vectorize the tokens into numerical vectors.

4.1.2 Model Training algorithm selection

Responsible for picking out the machine learning methods that is best suitable for the prepared features with its targets. Trains and optimizes the hyperparameters of the model.

4.1.3 Model prediction and evaluation

Responsible for transforming new textual data into the appropriate format, feeding newly pre-processed data into the trained model and evaluates the model's performance. Comparison of different model's techniques are also carried out here. Saving of the best model is the final step of the module.

4.2 Server Module

This module is in charge of loading the saved models gotten from the Training module's final step using Mlpack supported deserialization and serialization methods and serving the trained model for the web through defined routes and handles HTTP requests and respond with the classification results.

It utilizes Uwebsockets to create a responsive and efficient web server, listens for incoming requests and in turn is all containerized in a docker container to ensure consistent server environment.

4.3 Web server module

This module is responsible for providing the user-friendly interface that interacts with the server that hosts the machine learning model.

4.3.1 Frontend/View module

A module responsible for the presentation logic, how the data is presented to the end user and handles the end user's inputs and alters the view accordingly to user inputs. Tooling used in here is the ejs engine. For dynamic content generation.

4.3.2 Backend module

This module handles the backend logic of providing dynamic content generation, handling API request, interacting with databases.

This uses express to define the Api endpoints and also to communicate with the database to store or retrieve data and also generate dynamic html.

4.4 Database module

This module purpose is very straight forward. It stores and manages all forms of data based on the system requirement for the system use. Also helps to persists the data and provides it when readily needed.

5. Risk Analysis

- **Data quality and model accuracy**

  There is a huge risk on how good the data been feed into the model during training will be, as a model can always be as good as the data been feed to it.

  Mitigation: Regularly update and retrain the model with diverse datasets from different product reviews too.

- **Data security and privacy**

  The project handles user reviews, therefore there is a risk of sensitive information to be exposed. Measures of anonymizing data are be taken into consideration.

  Mitigation: Removal of personal identifiable information from the data before processing. The storage of data securely to prevent unauthorized access to the data. For this project the dataset source will not live in with the system. The dataset will be stored in a completely separate system that has not connection to the model system.

- **Scalability**

  This risk may only be gotten from big applications or systems that handle enormous amount of data per second. This risk was also taken into account with the use of C++ which is a very fast and robust language and the design of the system with scalability in mind

Mitigation: conducting stress testing and identify performances bottlenecks.

- **Dependencies mayhem**

  This is really a huge problem even when setting up the development environment. That is why I used docker dev containers to build and find the optimal environment that can run it efficiently

  Mitigation: Documenting the libraries and their dependency versions, building a docker image to hold a snapshot of the dev environment and production environment to run and build the system. Regularly check for updates and dependencies changes.

- **Deployment**

  There are risks during the deployment stage of the model which may require additional resources before it will be ready to serve. Failure to meet up with these requirements can lead to downtime, poor service. This risk can be mitigated with the use of docker for deployment.

  Mitigation: implement redundant server configurations, regularly server checkup and have a backup plan available.

6. Ethics

- **Bias**

  The model may be subject to unintentional biases residing in the data used to train it. Procedures of monitoring biases from the data source are also taken into consideration by regularly addressing and evaluating biases found in training data. [11]

- **Privacy**

  The system will have access to organizations customers or users reviews data. This can raise privacy concerns. The system already has data anonymization of the textual data. [12]

- **Transparency and Consent**

  The users of the system will be informed in full details how the data collected from them either using data source gathering or during predictions are been used and stored.

- **Accountability**

  It should be clear who is responsible for the model’s actions and outcomes. The task of correcting any harm caused by the model and the continuous update and monitoring of the model to always stay ethical.

- **Safety**

  İn terms of safety, the system will try it best to ensure it adopts and uses efficient and less power consuming cloud providers and software.

7. Conclusion

Due to its use of C++, this system exhibits expertise in memory management while working with large datasets, data pretreatment, natural language processing, docker usage, and web API development that set it apart from other projects. The goal of this project is to lower the cost of text classification in AI for developers and end users.

7.1 Benefits

Benefits to users

- In order to free up supplier time for other responsibilities, the technology will enable the automatic classification of review data.

- **Real-time feedback:** The system will be able to help business users to identify issues with their products from customers reviews more quickly and react swiftly.
- **Understanding user opinions:** Provide good in-depth knowledge of customer sentiment to help businesses improve their products and services, automating many labours intensive process of analysing text and saving time and resources.

Benefits to me

- *Skill improvements*: The project helps me further develop my skill in text vectorization and classification in a different language, system and development tools.
- *Portfolio improvement*: Can help boost my portfolio and showcase my ability in handling NLP tasks.
- *Problem solving skills improvement:* it introduces me to more complex libraries and compilation techniques, had to solve tasks such as compiling right and managing sizes, something that is abstracted in other modern languages.

Reason for choosing this project

I chose this project due to my previous knowledge of using Sklearn library in python to build a sentiment model with 82% accuracy result. Remaking this in C++ which doesn't have extensive support for NLP and few libraries for it was also a motivation as I want to broaden my knowledge in using languages and tools sets with virtually almost zero tutorials unlike with python or JavaScript where there are so many numerous tutorials for it.

7.2 Future Works

- İmprove on the number of data set used to train it
- Enable the feature of using it to analyse in real-time video chat
- Introduce more classification features that can be used to monitor brand trends in user chat forums, market research and competitive research
- Content and topic categorization, spam detection and malicious content detection.
- Be an addon to a speech to text system where it will analyse the text transformed by the speech to text system.

8. References

[1] B. Liu, "The Problem of Sentiment Analysis," Cambridge University Press eBooks, june 2015. [Online]. Available: https://www.cambridge.org/core/books/abs/sentiment-analysis/problem-of-sentiment-analysis/A0AFE2C49D72C5914C34DAC763BCD931. [Accessed 17 Dec 2023].

[2] Eden AI, "Eden AI Workflows," Edenai.co, 2023. [Online]. Available: “Eden AI Workflows,” Edenaihttps://www.edenai.co/workflows?referral=sentiment-analysis-vs-custom-text. [Accessed 17 Dec 2023].

[3] Levity, "Levity," Github.com, 04 Jul 2023. [Online]. Available: https://github.com/orgs/levityai/repositories?type=all. [Accessed 17 Dec 2023].

[4] Huggingface, "Huggingface," Huggingface.co, 11 Sep 2023. [Online]. Available: “huggingface (Hugging Face),” Huggingface.co, Sep. 11, 2023. https://huggingface.co/huggingface (accessed Dec. 22, 2023).. [Accessed 17 Dec 2023].

[5] typescriptlang, "JavaScript With Syntax For Types," typescriptlang, 2020. [Online]. Available: https://www.typescriptlang.org/. [Accessed 17 Dec 2023].

[6] R. R. C. e. al., "mlpack 4: a fast, header-only C++ machine learning library," *Journal of Open Source Software,* vol. vol 8 no. 82, no. R. R. Curtin et al., “mlpack 4: a fast, header-only C++ machine learning library,” Journdoi: https://doi.org/10.21105/joss.05026. url: https://arxiv.org/pdf/2302.00820.pdf, p. p. 5026, 2023.

[7] G. Ben-Evgi, "Exploring the Power of uWebSockets: A High-Performance WebSocket Library," Medium, 27 Jun 2023. [Online]. Available: https://ga1.medium.com/exploring-the-power-of-uwebsockets-a-high-performance-websocket-library-83aedc26ae1. [Accessed 17 Dec 2023].

[8] Wikipedia Contributors, "Adobe Illustrator," Wikipedia, 15 Dec 2023. [Online]. Available: https://en.wikipedia.org/wiki/Adobe_Illustrator. [Accessed 17 Dec 2023].

[9] Matplotlib, "Matplotlib — Visualization with Python," Matplotlib, 2023. [Online]. Available: https://matplotlib.org/. [Accessed 17 Dec 2023].

[10] Cmake.org, "CMake - Upgrade Your Software Build System," Cmake.org, 2023. [Online]. Available: https://cmake.org/. [Accessed 17 Dec 2023].

[11] S. M. Mohammad, "Ethics Sheet for Automatic Emotion Recognition and Sentiment Analysis," in *Computational Linguistics*, vol. vol 48, doi.org, 2022, p. pp. 239–278.

[12] S. M. Mohammad, "Ethics Sheet for Automatic Emotion Recognition and Sentiment Analysis," arXiv.org, 2021. [Online]. Available: https://arxiv.org/abs/2109.08256. [Accessed 17 Dec 2023].