Data-Driven Insights for Hate Speech Detection

This report provides an look at a data warehousing project aimed at detecting and analysing hate speech. The project was initiated due to the unavailability of a suitable Twitter API and involves the simulation of daily data analysis. The structure is similar to the final presentation.

Table of contents

Project Goals and Problem Statement

The project's objective is to develop a system capable of detecting hate speech in daily snapshots, tailored for NGOs and social justice organizations. A key challenge was the processing of extensive data without direct access to the Twitter API, necessitating the creation of a system to emulate real-time data flow. The challenge lay not only in processing a vast amount of data but also in devising a methodology to simulate daily data analysis in the absence of a direct Twitter API. This, as it was discovered later, was not available to public use for free.

Target User Profile

The end users are NGOs combating hate speech and cyberbullying, as well as humanoid advocacy groups and community and social justice organizations. The system is designed to meet their requirements for efficient and effective data analysis tools. These users require robust and intuitive tools to analyze large volumes of social media data for patterns and trends. The solution needed to be user-friendly, allowing non-technical users to glean insights and make data-driven decisions efficiently.

Datasets Overview

To overcome the limitations of not having direct Twitter API access, the project utilized datasets from HugginFace. These datasets were processed in batches, simulating a real-time environment. The data collection process was meticulously planned to include data from CSV files, ensuring that no duplication occurred in the database. The first preprocessing stage was not that critical for current state of project, focusing on retaining only English tweets and discarding any irrelevant attributes to maintain data quality and relevance. It can be omitted and the database could be changed. This flexible strategy is about setting up our database for a variety of sources that we might tap into later on. It's also about not being tied down by the API and making sure we can get to the raw data as it happens.

System Architecture and Tools

The project's system architecture is robust, encompassing several key components for efficient data handling and analysis. It includes:

- **Data collector and basic preprocessor**: Utilizing Python scripts, this component is crucial for simulating real-time data flow and preprocessing data. It ensures that only relevant information, like English tweets, is processed and stored.
- **MySQL Database**: The database is at the core of the system, managing raw data efficiently as well as the star schema design. It is optimized for scalability and future integration of various data sources.
- **AI Model**: The AI model, designed for a regression task, categorizes speech into hate speech, neutral, or counter-speech. An additional disrespect score, using the same dataset, adds depth to the analysis.
- **Power BI**: A pivotal tool in the project, Power BI is used for creating dynamic and interactive dashboards. These dashboards are crucial for visualizing data trends and providing insightful reports. Power BI's capability to handle large datasets and its intuitive user interface make it an ideal choice for presenting the hate speech and respect scores in a clear and actionable format. It allows users to easily interpret complex data patterns, such as the observed increase in hate speech during evening hours.

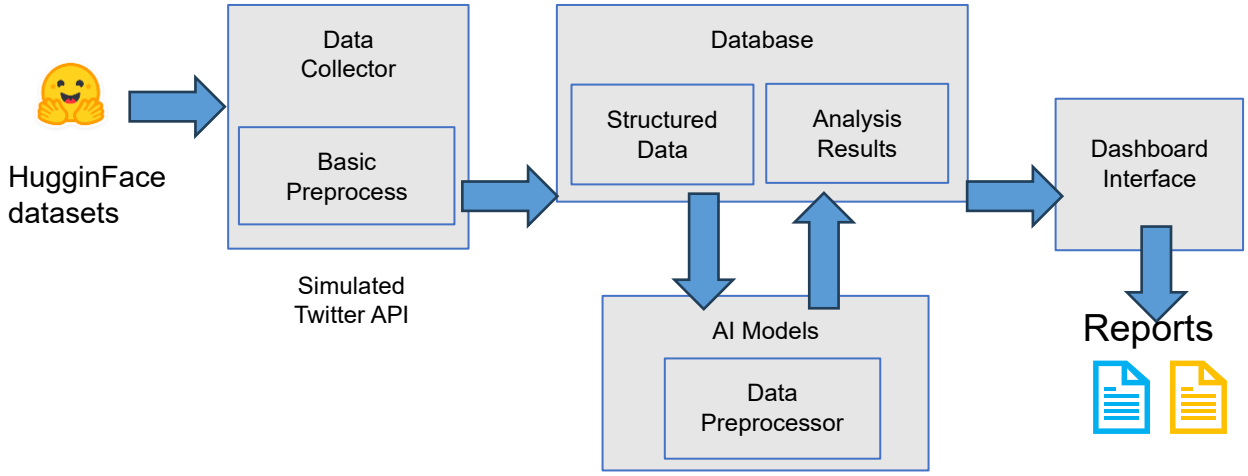

Figure 1 – Simplified system architecture

AI Model for Hate Speech Prediction

The AI model, at the heart of this project, was designed to perform a regression task. It generated a continuous hate speech score, along with an additional respect score. These scores were interpreted on a scale, allowing for nuanced categorization of the speech into hate speech, neutral, or counter-speech, as well as measuring varying levels of respect. This granular approach to scoring provided a comprehensive understanding of the nature and severity of online interactions.

Data Storage

	Table	Action							Rekordy	Typ	Collation	Rozmiar	Nadmiar
☐	content_dim	☆	Browse	Structure	Szukaj	Wstaw	Empty	Drop	845	InnoDB	utf8mb3_general_ci	48.0 KB	-
☐	time_dim	☆	Browse	Structure	Szukaj	Wstaw	Empty	Drop	845	InnoDB	utf8mb3_general_ci	64.0 KB	-
☐	tweet_fact	☆	Browse	Structure	Szukaj	Wstaw	Empty	Drop	845	InnoDB	utf8mb3_general_ci	112.0 KB	-
☐	tweet_raw	☆	Browse	Structure	Szukaj	Wstaw	Empty	Drop	13,635	InnoDB	utf8mb3_general_ci	3.5 MB	-
	4 tables	Suma							16,170	InnoDB	utf8mb3_general_ci	3.7 MB	0 B

Figure 2 – Tables in the database

At the heart of this data warehousing project lies a meticulously designed database, crafted to transform the raw data into actionable insights. It is made for both raw data and star schema in the same time. Let's unveil the layers:

tweet_raw: This table is the storage to the raw data of tweets as they come in. It includes:

- *tweet_raw_id*: The unique identifier for each tweet record.
- *tweet*: The text content of the tweet itself.
- *likes*: The number of likes the tweet has received.
- *replies*: The number of replies to the tweet.
- *retweets*: The number of times the tweet has been retweeted.
- *quotes*: The number of times the tweet has been quoted.
- *creation_date*: The date and time the tweet was created.

content_dim: This dimension table includes information about the content of tweets. It is used for analyzing tweets based on their content characteristics. It includes:

- *content_dim_id*: A unique identifier for each content dimension record.
- *popularity_level*: A classification of how popular the tweet; based on likes; structure: Low, Moderate, High.
- *engagement_level*: A classification of how engaging the tweet is; based on replies; structure: Low, Moderate, High.
- *distribution_level*: How widely the tweet has been shared or distributed; connected with number of retweets; structure: Low, Moderate, High.
- *mention_level*: The level of mentions of other users in the tweet; connected with quotes; Structure: Low, Moderate, High.
- *content_length*: The length of the tweet's content; structure: Short, Medium Long.

time_dim: This table is likely used to allow for analysis over various time dimensions. It includes:

- *time_dim_id*: A unique identifier for each time dimension record.
- *year*: The year the tweet was created.
- *month*: The month the tweet was created.
- *day*: The day of the month the tweet was created.
- *day_time*: The time of day the tweet was created: Morning, Afternoon, Evening.
- weekday: The day of the week the tweet was created.

tweet_fact: This is a fact table that likely serves as the central table for the star schema, connecting the dimensions together for comprehensive analysis. It includes:

- *content_dim_id*: A reference to the content_dim table.
- *time_dim_id*: A reference to the time_dim table.
- *hate_speech_score*: A numerical score indicating the likelihood that the tweet contains hate speech.
- *respect_score*: A numerical score indicating the level of respect or disrespect in the tweet.

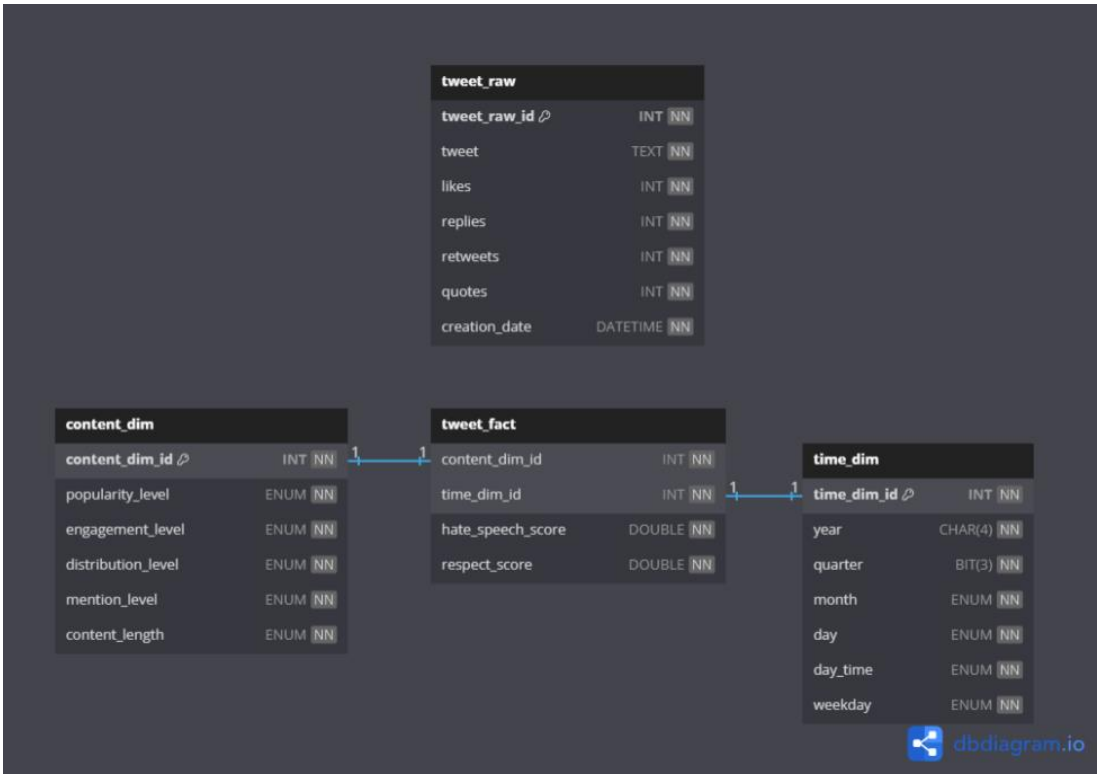

Figure 3 – Database schema

Dashboard and Reporting

Two reports were prepared: one for the hate speech score and the other for the respect score.

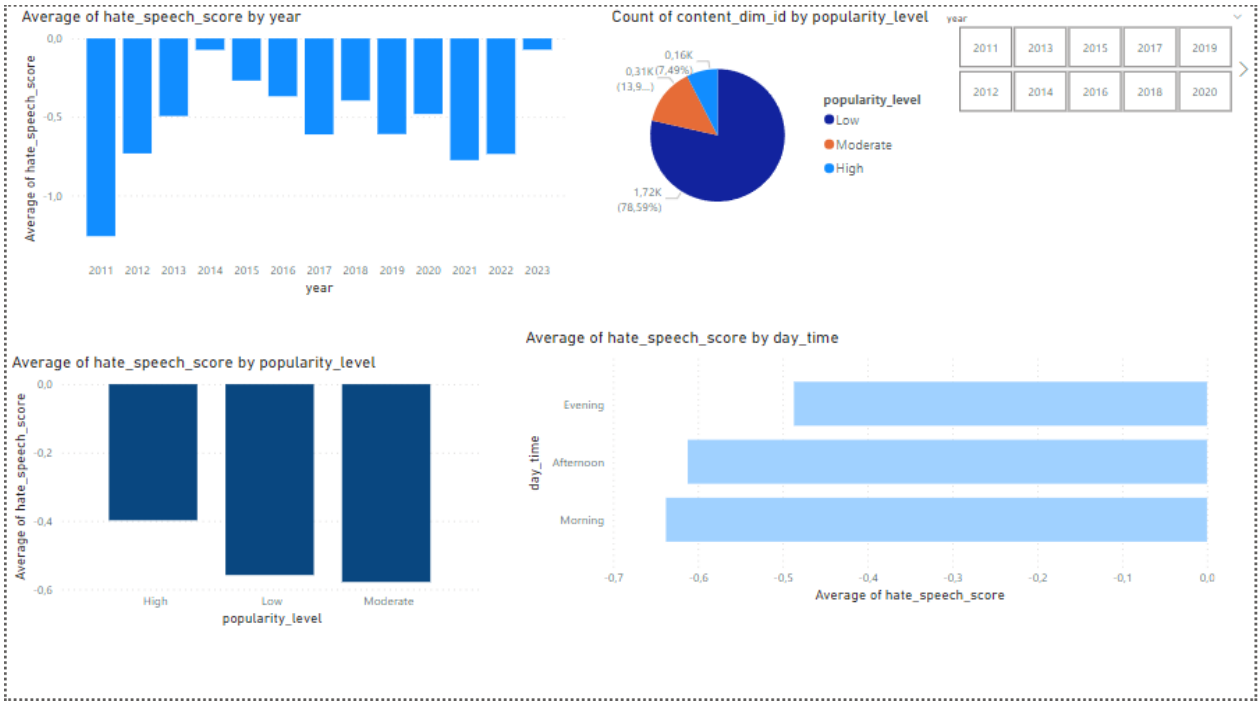

Figure 4 – Hate speech dashboard

The dashboard for hate speech score is divided into four primary sections:

1. **Average of Hate Speech Score by Year**: This bar chart tracks the average hate speech score over the years, providing a clear view of how the incidence or intensity of hate speech has evolved annually.
2. **Count of Content Dimension ID by Popularity Level**: The pie chart presents the distribution of tweets based on their popularity levels over selected years, offering insight into which tweets, categorized by their reach and engagement, dominate the dataset.
3. **Average of Hate Speech Score by Popularity Level**: This bar graph compares the average hate speech scores across different popularity levels of tweets, shedding light on whether the reach of a tweet correlates with the amount of hate speech it contains.
4. **Average of Hate Speech Score by Day Time**: The final bar chart shows the average hate speech score segmented by the time of day, highlighting potential patterns in hate speech occurrence during different periods such as morning, afternoon, and evening.

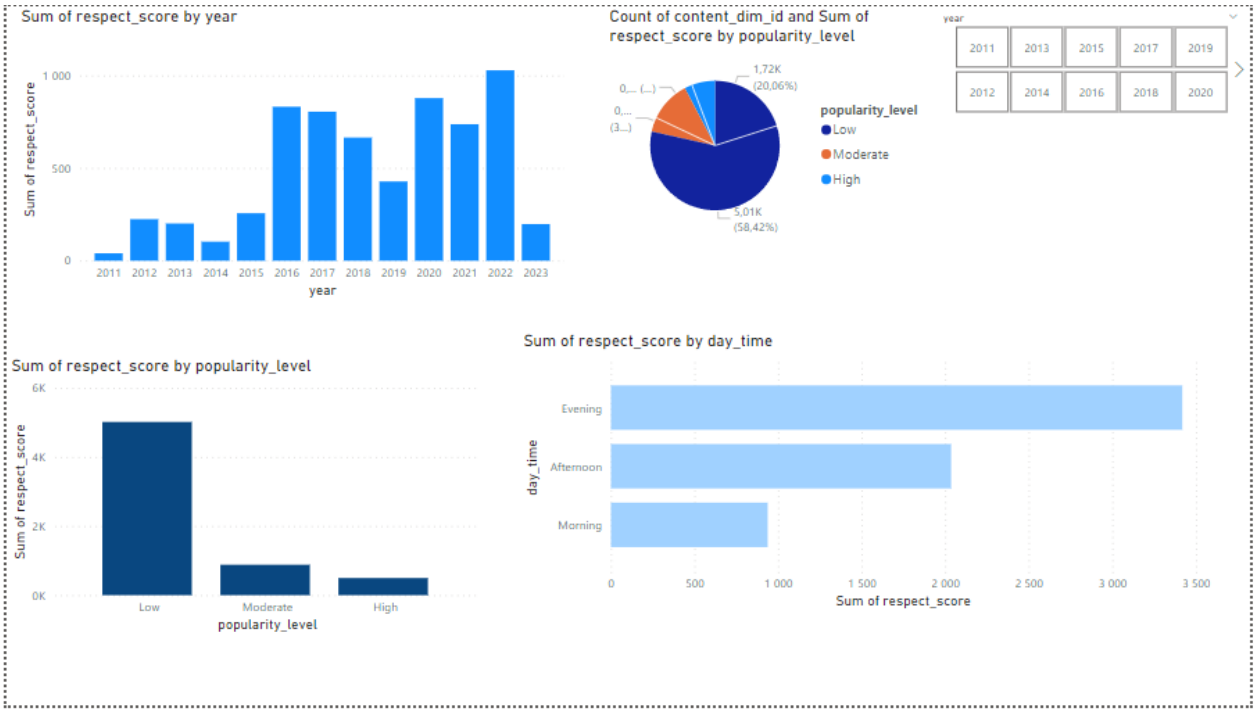

Figure 5 – Respect score dashboard

For the respect score dashboard, similar methods of visualization was provided. Analysis of the data revealed that hate speech and disrespect tend to increase during evening hours.

Project Development Timeline

The project was developed from scratch, involving the creation of Python scripts to simulate an API, setting up a MySQL database, and developing an AI model for real-time analysis. Most of the plan was according to the planned timeline form the first presentation.

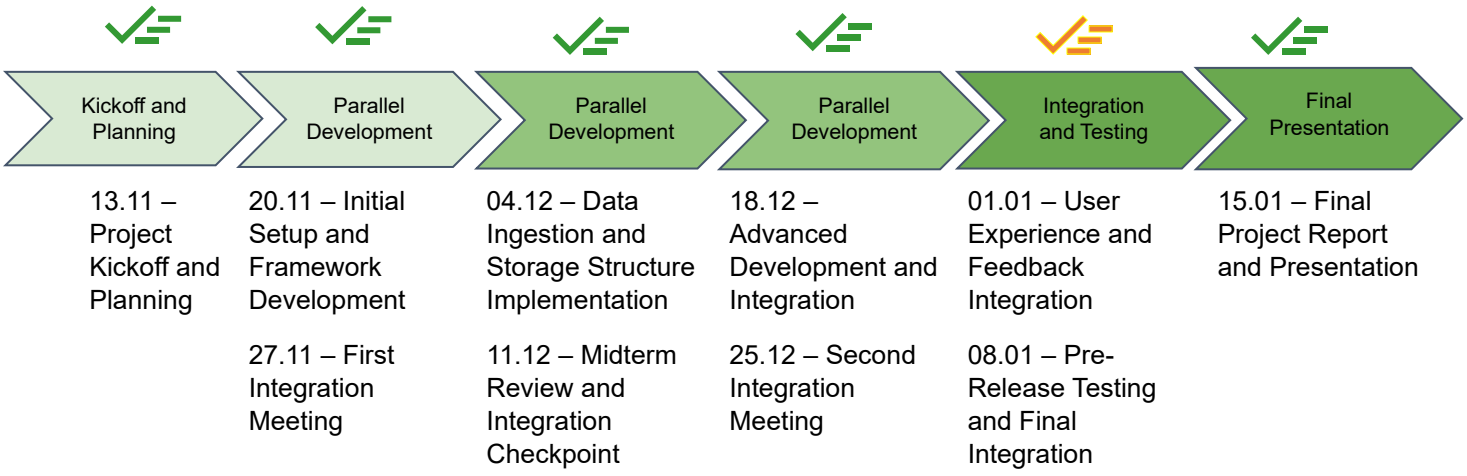

Figure 6 – Project timeline

Conclusion

This project demonstrates an innovative approach to using data warehousing and AI for social media monitoring, offering a scalable and insightful solution for detecting and analysing hate speech. The completion of this project marks a significant milestone in the use of data warehousing and AI for monitoring social media.

Table of Figures: