# CCT College Dublin

Assessment Cover Page

To be provided separately as a word doc for students to include with every submission

|  |  |
| --- | --- |
| Module Title: | AdvData, Big Data |
| Assessment Title: | CA2 Integrated with Big Data |
| Lecturer Name: | David McQuaid, Muhammad Iqbal |
| Student Full Name: | Farhad KHANKISHIYEV |
| Student Number: | 2023068 |
| Assessment Due Date: | 17.11.2023 |
| Date of Submission: | 22.11.2023 |

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

GITHUB Link: (<https://github.com/FarhadKhankishiyev068/2023068_BIGData_CA2>)

*TIME SERIES ANALYSIS FOR TWEETS*

INTRODUCTION:

In recent years, the integration of natural language processing and machine learning techniques has become pivotal in extracting valuable insights from vast datasets. This academic exploration delves into the utilization of PySpark and PyTorch to analyze and model sentiment trends from a Twitter dataset. The rationale behind this choice lies in the robust capabilities of PySpark for distributed data processing, coupled with PyTorch's flexibility for implementing deep learning models.

Data Processing and Storage:

The initial stages of this project involve leveraging PySpark to preprocess and aggregate sentiment data. PySpark's ability to handle large-scale distributed data seamlessly aligns with the diverse and dynamic nature of Twitter data. Aggregated data is stored efficiently, ensuring accessibility for subsequent analytical tasks.

Programming Language Choice:

The decision to employ Python as the primary programming language is grounded in its versatility, extensive libraries, and community support. Python serves as an ideal medium for seamless integration between PySpark and PyTorch, fostering a cohesive workflow from data processing to machine learning model development.

Machine Learning Models and Algorithms:

The focus then shifts to sentiment analysis, where PyTorch facilitates the implementation of a Long Short-Term Memory (LSTM) network for time series forecasting. The LSTM model is chosen due to its efficacy in capturing temporal dependencies within sequential data, aligning with the time-sensitive nature of sentiment trends.

As we proceed, the subsequent sections will delve deeper into the implementation details, discussing the methodology, results, and implications of the chosen approach. This academic endeavor aims to showcase the synergy between PySpark and PyTorch in extracting meaningful insights from Twitter sentiment data.

Objective:

The primary objective of this project is to create an advanced sentiment analysis system using cutting-edge technologies like PySpark and PyTorch. Our goal is to accurately analyze public sentiment on Twitter, providing valuable insights into trends and patterns over time.

Problem Statement:

In the age of social media dominance, understanding public sentiment on platforms like Twitter is crucial. However, the vast and dynamic nature of Twitter data poses challenges for sentiment analysis. This project addresses these challenges by implementing a robust solution that involves efficient data processing, aggregation, and the development of a sophisticated sentiment analysis model. The aim is to contribute to the advancement of sentiment analysis techniques with practical applications in real-world scenarios.

LITERATURE REVIEW AND RESEARCH

2.1The text explores sentiment analysis, a technique vital in social sciences, computer science, and psychology. It highlights the challenge of analyzing large datasets and the importance of understanding language for effective sentiment extraction. The study aims to compare sentiment analysis libraries and machine learning algorithms in Python and R, focusing on comment polarity classification accuracy. The methodology involves a literature review and a novel approach using advanced machine learning methods. The expected results aim to contribute to sentiment analysis advancement and guide future research and applications.(“COMPARATIVE ANALYSIS OF LIBRARIES FOR THE SENTIMENTAL ANALYSIS,” n.d.)

2.2 The paper introduces a framework for multimodal sentiment analysis and emotion recognition. By employing convolutional neural networks for feature extraction from text and visual data, the proposed approach demonstrates a 10% performance enhancement compared to existing methods. Notably, the paper emphasizes the significance of considering speaker independence, modality importance, and generalizability in multimodal sentiment analysis research—a factor often overlooked. The work sets a new benchmark for future research in this field, shedding light on crucial aspects and providing a comprehensive perspective for effective sentiment analysis across different modalities.(“Benchmarking Multimodal Sentiment Analysis,” n.d.)

2.3 The article proposes an innovative method for sentence-level sentiment analysis, addressing challenges in capturing sentiment nuances within individual sentences. It introduces expressive constraints derived from lexical and discourse knowledge, enhancing Conditional Random Field (CRF) models through posterior regularization. The approach demonstrates superior performance in both supervised and semi-supervised settings, emphasizing the valuable role of discourse information in sentiment analysis. The incorporation of long-distance discourse relations and the semi-supervised nature of the method make it a noteworthy contribution to the field. The use of posterior regularization with linguistically-motivated constraints represents a novel and effective strategy, highlighting the importance of context in disambiguating sentence-level sentiment.(“Context-aware Learning for Sentence-level Sentiment Analysis with Posterior Regularization,” n.d.)

2.4 Sentiment Analysis (SA) has become a crucial aspect of Natural Language Processing (NLP), automating the detection and understanding of emotions in written text. This study provides a comprehensive review of the latest trends and techniques in SA, covering various methods such as lexicon-based, graph-based, machine learning, and deep learning. The paper addresses challenges like sarcasm, irony, and ethical concerns, using Twitter as a case study. It explores applications in social media, healthcare, marketing, finance, and politics, offering a comparative analysis of existing trends, techniques, datasets, and evaluation metrics. The goal is to guide researchers and practitioners by identifying gaps and suggesting improvements in SA processes for more efficient and accurate outcomes.(“Exploring Sentiment Analysis Techniques in Natural Language Processing: A Comprehensive Review,” n.d.)

2.5 The paper introduces Convolutional Neural Networks (CNNs) for sentiment classification, emphasizing the growing importance of analyzing textual data automatically. As textual data continues to increase, sentiment analysis becomes crucial for various applications, such as predicting sentiment categories like positive or negative. The authors propose a CNN architecture, specifically employing consecutive convolutional layers, demonstrating its effectiveness for longer texts compared to other state-of-the-art deep learning models. The study aims to contribute to sentiment analysis, particularly in the context of movie reviews and customer feedback.(“Sentiment Classification Using Convolutional Neural Networks,” n.d.)

2.6 This study investigates the impact of stay-at-home orders during the COVID-19 pandemic on Twitter sentiment in the United States. Analyzing 7.4 billion geo-tagged tweets, the research employs a multiperiod difference-in-differences regression model to examine sentiment changes before and after the implementation of stay-at-home orders across 51 states and territories. The study focuses on feedback from various groups, emphasizing vulnerable populations such as older individuals with underlying health conditions, small and medium enterprises, and low-income groups. Results indicate a positive shift in public sentiment after the implementation of stay-at-home orders, particularly benefiting states with fewer vulnerable groups. However, this positive sentiment diminishes over time, with negative sentiment more likely in states with a higher proportion of vulnerable populations. The study highlights the importance of considering economic and demographic factors in understanding the dynamics of public opinion during a pandemic.(“Staying Home, Tweeting Hope: Mixed Methods Study of Twitter Sentiment Geographical Index During US Stay-At-Home Orders,” n.d.)

2.7 Limited Representativeness : The study emphasizes the need to scrutinize social media user demographics for a nuanced understanding of dynamics. Despite Facebook's global dominance, Twitter, with its accessible API, is a key research focus, albeit with a skewed user demographic. The concentration of users in specific age groups and regions raises concerns about representativeness, and the filter bubble hypothesis underscores the potential polarization of social networks.

Misinformation Challenges : The impact of misinformation, especially during crises like the COVID-19 pandemic, is highlighted. False rumors on social media can disrupt societal functions, and the study identifies various forms of misinformation. The rapid spread of false information, even surpassing accurate content, poses significant challenges. The study advocates for automated analysis systems incorporating machine learning to identify and discard misinformation, considering contextual factors for improved accuracy.(“The Ethical Risks of Analyzing Crisis Events on Social Media with Machine Learning,” n.d.)

2.8 Leveraging Sentiment Analysis for Patient Experience Evaluation in Health Care

Background:

The research explores the untapped wealth of unstructured health care information available online, specifically in patient comments. It employs sentiment analysis, a machine learning technique, to categorize these free-text comments into positive or negative sentiments. The primary goal is to predict patient recommendations, cleanliness assessments, and perceptions of dignity, and subsequently compare these predictions with patients' quantitative ratings.

Methods:

The study utilizes machine learning techniques, applying sentiment analysis to 6,412 online comments about hospitals from the English National Health Service (NHS) website in 2010. Weka data-mining software is employed for sentiment analysis. The outcomes of sentiment analysis are then compared with both quantitative patient ratings and national patient survey data for 161 acute adult hospital trusts in England.

Results:

The findings reveal a noteworthy agreement (81-89%) between sentiment analysis-derived predictions and patients' quantitative ratings for cleanliness, dignity, and overall recommendation. Moreover, the machine learning predictions exhibit mild to moderate associations with responses from the national patient survey.

Conclusions:

The study concludes that sentiment analysis proves to be a promising tool for extracting valuable insights from patients' unstructured comments. The accuracy of predictions highlights the potential of machine learning approaches in comprehending and forecasting patients' opinions, offering a unique perspective that complements traditional survey methods.(“Use of Sentiment Analysis for Capturing Patient Experience From Free-Text Comments Posted Online,” n.d.)

CRITICAL EVALUATION

Comprehensive Rationale for Methodological Choices

In elucidating the strategic decisions woven into the fabric of our project, a nuanced evaluation unveils the rationale and foresight embedded in our methodological choices, providing a comprehensive understanding of the journey from conceptualization to execution.

1. Data Processing and Storage:

Strategic Advantages:

Scalability Mastery: The adoption of PySpark for data processing was no arbitrary decision. Its distributed computing prowess was a strategic move, ensuring our ability to seamlessly scale our analyses amidst the relentless influx of Twitter data.

Agile Responsiveness: PySpark, seamlessly integrated with Python, empowered us to swiftly respond to the dynamic landscape of Twitter. The real-time adaptability to emerging trends was a game-changer in staying ahead in the fast-paced realm of social media analytics.

Strategic Considerations:

Learning Dynamics: The decision to embrace PySpark wasn't without its considerations. While it brings robust capabilities, the learning curve demands attention. Our team navigated this challenge, emphasizing knowledge transfer to optimize the use of PySpark.

2. Programming Language Choice:

Strategic Advantages:

Interdisciplinary Harmony: The selection of Python as our primary language wasn't just about syntax; it was a deliberate choice to foster interdisciplinary collaboration. The readability of Python code facilitated seamless communication within our diverse team, promoting a harmonious workflow.

Ecosystem Synergy: Python's vast ecosystem was harnessed strategically. The integration of diverse libraries expedited our tasks, showcasing our commitment to an efficient and resourceful development process.

Strategic Considerations:

Performance Scrutiny: Python's versatility comes with performance considerations. We approached this with a strategic mindset, addressing potential performance bottlenecks through careful optimization and, if required, integrating performance-focused languages.

3. Machine Learning Models:

Strategic Advantages:

Temporal Acuity: Opting for LSTMs was a testament to our understanding of the temporal intricacies within Twitter data. These models, with their ability to capture long-term dependencies, were strategically aligned with the temporal nuances crucial for accurate sentiment analysis.

Deep Learning Precision: Leveraging deep learning capabilities, LSTMs proved instrumental in comprehending the complex linguistic patterns pervasive in Twitter language, elevating the precision of our sentiment analysis.

Strategic Considerations:

Resource Investment: LSTMs, being computationally intensive, demanded a strategic investment in computational resources. This foresight ensured our ability to harness the full potential of these deep learning models.

4. Sentiment Analysis Algorithm:

Strategic Advantages:

Contextual Mastery: Choosing LSTM for sentiment analysis underscored our commitment to contextual precision. The sequential processing prowess of LSTMs resonated with the contextual intricacies inherent in Twitter language.

Training Adaptability: LSTMs' iterative nature played strategically into our hands. The adaptability of these models over time was a crucial strength, aligning seamlessly with the dynamic landscape of Twitter sentiment.

Strategic Considerations:

Interpretability Balance: Acknowledging the "black-box" nature of deep learning models, including LSTMs, we strategically balanced accuracy with interpretability. This consideration is vital, especially when transparency in model decision-making is paramount.

Synthesis of Strategic Choices:

The amalgamation of PySpark for data processing, Python for its interdisciplinary synergy, LSTMs for deep learning capabilities, and the chosen sentiment analysis algorithm represents a strategically crafted framework for dissecting

Twitter sentiment. Each choice serves as a strategic pillar, contributing unique strengths while navigating the associated considerations. The success of our project hinges on this intricate interplay, and our commitment to ongoing refinement positions us strategically in the ever-evolving landscape of Twitter data and sentiment dynamics.

CONCLUSION

In embarking on this project, my aim was to explore the synergy between PySpark and PyTorch for sentiment analysis on Twitter data, recognizing the unique strengths each technology offers.

The utilization of PySpark for data processing proved strategic, given its scalability for handling large-scale distributed data. Navigating the learning curve associated with PySpark was a challenge, but the real-time adaptability to emerging trends in the dynamic Twitter landscape proved transformative.

Python, chosen as the primary programming language, served as a facilitator of interdisciplinary collaboration. Its vast ecosystem streamlined tasks and, despite performance considerations, contributed to an efficient development process through careful optimization.

The core of the project involved implementing a Long Short-Term Memory (LSTM) network with PyTorch for sentiment analysis. This decision was rooted in the efficacy of LSTMs in capturing temporal dependencies within sequential data, aligning well with the time-sensitive nature of sentiment trends. However, the computationally intensive nature of LSTMs required a strategic investment in computational resources.

The sentiment analysis algorithm, particularly the LSTM implementation, emphasized contextual precision while balancing the interpretability challenge inherent in deep learning models. The strategic choices made in this project—PySpark for scalability, Python for interdisciplinary harmony, LSTMs for temporal acuity—formed a meticulously crafted framework for dissecting Twitter sentiment.

In conclusion, this project goes beyond technological integration; it reflects a journey of resilience and strategic decision-making. It positions me to navigate the ever-evolving landscape of data and sentiment dynamics. This experience has been transformative, shaping not only my technical skills but also my approach to complex problem-solving in the dynamic field of data science.

Challenges and Difficulties

Throughout the course of this project, I encountered several challenges that added layers of complexity to the implementation. These challenges, while demanding, provided invaluable learning experiences and opportunities for growth.

PyTorch Implementation Challenges:

Implementing Long Short-Term Memory (LSTM) networks with PyTorch, although powerful, presented challenges. The computational intensity of deep learning models like LSTMs required a substantial investment in computational resources. This posed constraints on the scale of data that could be effectively processed within available infrastructure.

Data Size Challenges in PySpark:

PySpark's capacity to seamlessly handle large-scale distributed data is commendable. However, as the Twitter dataset scaled, I encountered difficulties in efficiently processing and analyzing vast amounts of data. Optimizing the code for performance became crucial, and at times, limitations in the available computing resources impacted the execution of time series analyses.

Interplay of PySpark and PyTorch:

Integrating PySpark for data processing and PyTorch for deep learning introduced additional complexity. Coordinating the flow of data between these technologies, especially when dealing with extensive datasets, required careful orchestration. Balancing the computational demands of PyTorch with the distributed processing capabilities of PySpark demanded strategic optimizations.

Code Execution Challenges:

Running time series analyses, particularly those involving PyTorch, occasionally led to code execution challenges. Issues related to memory management and code optimization became prominent, affecting the reproducibility and efficiency of the analyses.

Learning Curve with PySpark:

Embracing PySpark for distributed data processing was a strategic choice, but it came with a learning curve. Navigating this learning curve, especially in the context of real-time adaptability to the dynamic Twitter landscape, posed a significant challenge. Knowledge transfer within the team was crucial to optimize the use of PySpark effectively.

Despite these challenges, each obstacle became an opportunity for problem-solving and skill development. The iterative nature of addressing these difficulties not only refined the project's implementation but also enhanced my proficiency in handling real-world complexities in data science and machine learning projects.

V REFERENCES

[1] Erik Cambria a , Devamanyu Hazarika d , Soujanya Poria c , Amir Hussain b , R.B.V. Subramaanyam d

[2] Faculty of Statistic and Computer Engineering Universidad Nacional del Altiplano de Puno, P.O. Box 291 Puno - Peru

[3] Bishan Yang Department of Computer Science Cornell University, Claire Cardie Department of Computer Science Cornell University

[4] Karthick Prasad Gunasekaran1 College of Information and Computer Sciences, University of Massachusetts, Amherst, United States

[5] Hannah Kim Received: 29 April 2019 / Revised: 22 May 2019 / Accepted: 4 June 2019 / Published: 7 June 2019

[6] Xinming Xia 1 2 3, Yi Zhang 4 5, Wenting Jiang 6, Connor Yuhao Wu 6Affiliations expandPMID: 37486758 PMCID: PMC10407645 DOI: 10.2196/45757

[7] Angelie Kraft1,2,∗, Ricardo Usbeck1,21Universität Hamburg, Department of Informatics, Vogt-Kölln-Straße 30, 22527 Hamburg2Hamburger Informatik Technologie-Center e.V. (HITeC), Vogt-Kölln-Straße 30, 22527 Hamburg

[8] Monitoring Editor: Gunther EysenbachReviewed by Marina Sokolova, Elad Yom-Tov, Phillip Brooker, and Andreas HolzingerFelix Greaves, MBChB,corresponding author12 Daniel Ramirez-Cano, PhD,2 Christopher Millett, PhD,1 Ara Darzi, MD,2 and Liam Donaldson, MD21Department of Primary Care and Public Health, Imperial College London, London, United Kingdom2Centre for Health Policy, Imperial College London, London, United KingdomFelix Greaves, Department of Primary Care and Public Health, Imperial College London, Charing Cross Hospital, London, W6 8RF, United Kingdom,