Capstone Project

Harnessing NLP to Detect Stress in Social Media

Early Intervention for Mental Wellbeing

Presenter: Jimmy Chong

Agenda

- Bio
- Project Context & Business Problem
- Business & Data Science Considerations
- Data Overview
- Data Exploration
- Data Split
- Data Overview
- Deliver
- Summary, Conclusions & Next Steps
- Appendix

Bio

- **Solution Architect** with experience in software design and project management
- **Transitioning to Data Science** through a Certificate in Data Science/AI from the Institute of Data
- **Skills:** Data analysis, machine learning, project management, programming (Python, C#, VB.NET), databases
- **Motivation:** Leverage technical background and data science skills to solve real-world problems and drive innovation

Key Projects:

- Stress Detection: Developing an NLP-based model to detect stress in social media text
- Predicting and Understanding Factors that Influence Hotel Booking Cancellations
- Predicting West Nile Virus Presence Using Machine Learning

Project Context & Business Problem

- **Industry:** NLP and Mental Health
- **Problem:** Detecting stress in social media text
- **Interest:** Growing mental health concerns, potential for early intervention, NLP advancements
- **Previous Work:**
  - **NLP techniques:** Sentiment analysis, topic modeling, machine learning
  - **Domains:** Twitter, Reddit, general datasets
  - **Key findings:** Promising accuracy, varying generalizability and robustness
  - **Limitations:** Reliance on labeled datasets, potential for bias
  - **Contribution:** Exploring LSTM networks, addressing limitations

Business & Data Science Considerations

- **Stakeholders:** Mental health professionals, social media platforms, technology companies, individuals
- **Business Question:** Can we accurately detect stress in social media text?
- **Business Value:** Early intervention, improved user experience, new market opportunities
- **Data:**
  - **Question:** Can we extract effective features for stress prediction?
  - **Required:** Social media text, stress labels, metadata
  - **Sourced:** Reddit, Twitter, Kaggle
  - **Generation:** User-generated content
  - **Future Sourcing:** Continued access to APIs, exploring other platforms

Data Overview (1)

Stress Detection from Social Media Articles

- **Source:** Kaggle Dataset
- **Objective:** Develop a more accurate and efficient NLP-based approach to automatically detect stress in social media text, enabling early intervention and improved mental health support.

Data Overview (2)

Stress Detection from Social Media Articles

- **Datasets:**
  - Constructed four datasets using text articles from Reddit and Twitter.
  - Each article is labeled with a class value of '0' (Stress Negative) or '1' (Stress Positive).
- **Dataset Descriptions:**
  - **Reddit Combi:** This dataset combines title and body text from stress and non-stress related subreddits.
  - **Reddit Title:** This dataset consists of titles from stress and non-stress related subreddits.
  - **Twitter Full:** This dataset contains stress and non-stress related tweets.
  - **Twitter Non-Advert:** This dataset is a denoised version of Twitter Full, removing potential advertisements.

Data Exploration (1)

Understanding the Data

- Explored the data using Python libraries like pandas to understand the number of rows, column names, and data types.
- Examined the first few rows of each Data Frame to get a sense of the content and labels.
- Checked for missing values and handled them by dropping rows with missing entries (dropna).
- Preprocessed the data:
  - Dropped unnecessary columns (Reddit Combi).
  - Cleaned hashtags in Twitter data using regular expressions.
  - Concatenated all preprocessed Data Frames into a single one.
- Analyzed the data distribution:
  - Visualized the distribution of stress labels using a bar chart.
  - Examined the distribution of text length for stress and non-stress articles using a histogram.
  - Generated word clouds to visualize frequently used words in stress and non-stress articles.

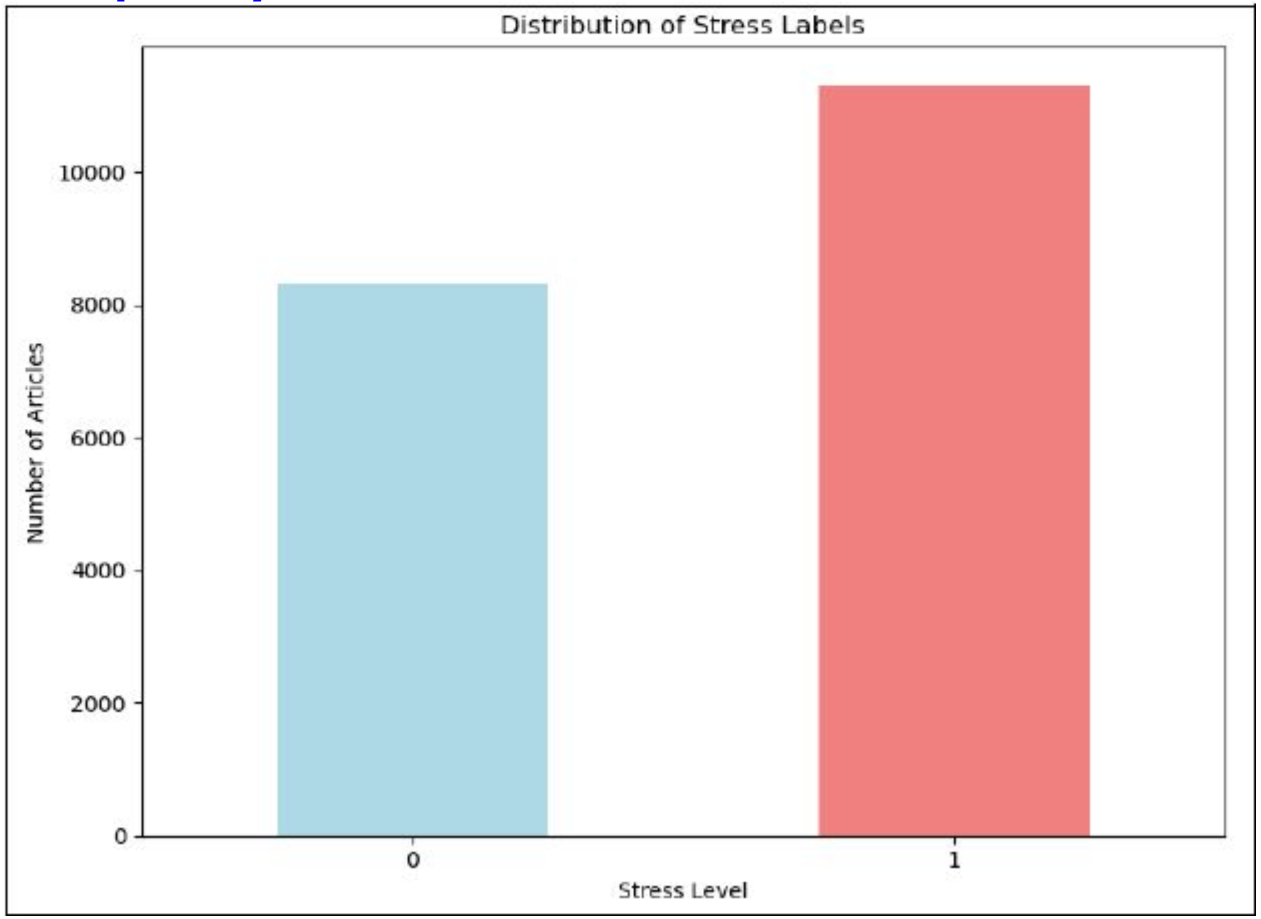
Data Exploration (2)

Understanding the Data

- **Number of Rows and Columns:**
  - Reddit Combi: 3123 rows, 4 columns
  - Reddit Title: 5556 rows, 2 columns
  - Twitter Full: 8900 rows, 3 columns
  - Twitter Non-Advert: 2051 rows, 2 columns
- **Data Types:**
  - **title**: object (text)
  - **label**: int64/boolean (stress label)
  - **body**: object (text) (only in Reddit Combi)
  - **hashtags**: object (text) (only in Twitter Full)
- **Missing Values:**
  - Handled missing values by dropping rows with missing entries in body column.
- **Data Preprocessing:**
  - Dropped unnecessary columns in Reddit Combi.
  - Cleaned hashtags in Twitter Full using regular expressions.
  - Concatenated all datasets into a single DataFrame.

Data Exploration (3)

Frequency of Stressful and Non-Stressful Posts

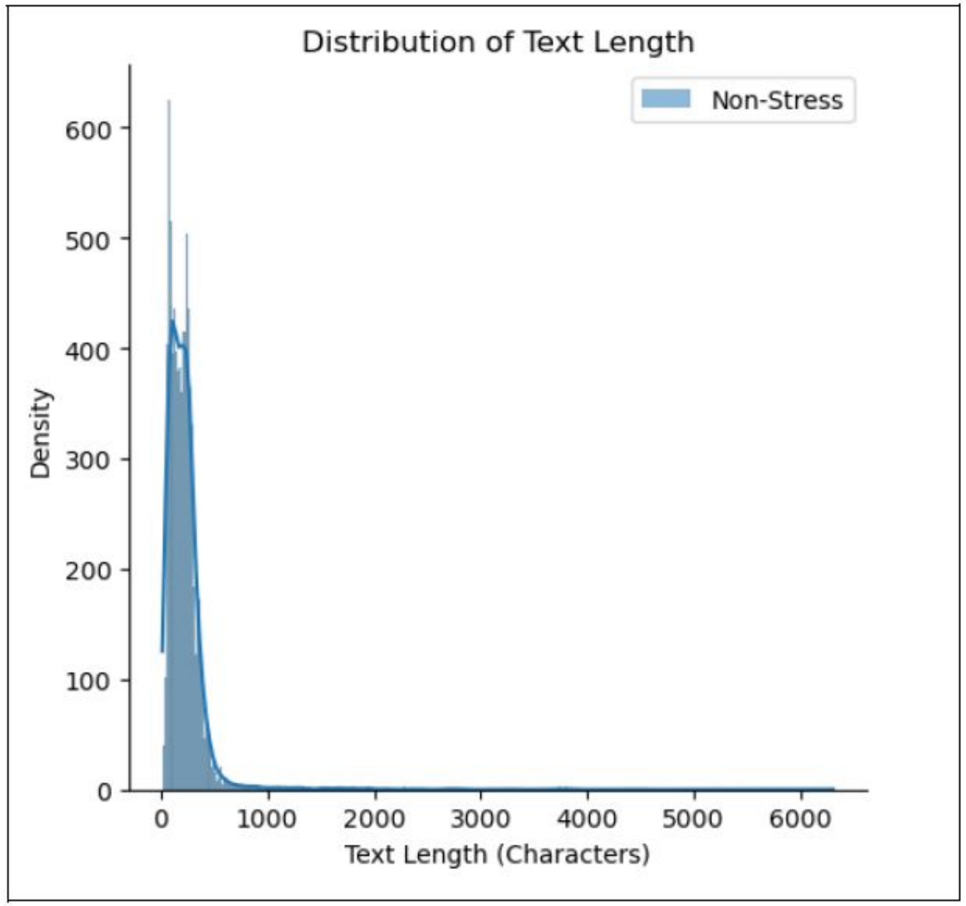


Key Notes:

- **Class imbalance:** More non-stressful articles than stressful ones.
- **Impact on modeling:** Require techniques like class weighting or oversampling.
- **Further investigation:** Explore factors contributing to imbalance (e.g., labeling difficulty, data collection bias).

Data Exploration (4)

Distribution of Text Length



Key Notes:

- **Skewed distribution:** Most posts are relatively short.
- **Overlapping distributions:** Text length alone may not be a strong predictor.
- **Non-stressful posts:** Slightly longer on average

Data Exploration (5)

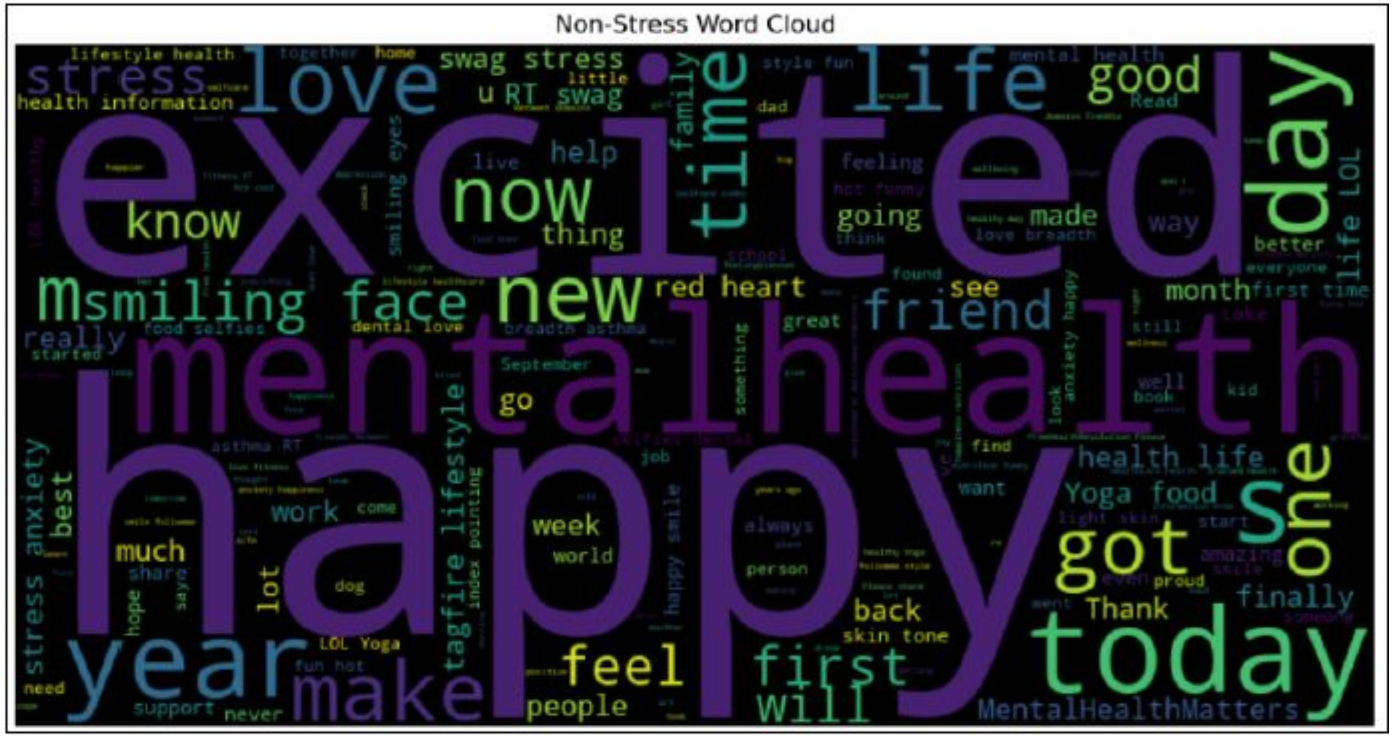
Common Words Associated with Stress

Key Notes:

- **Negative emotions:** "stress," "depressed," "sad," "anxious"
- **Life challenges:** "work," "school," "relationships," "personal problems"
- **Seeking help:** "help," "support," "therapy"

Data Exploration (6)

Common Words Associated with Non-Stress



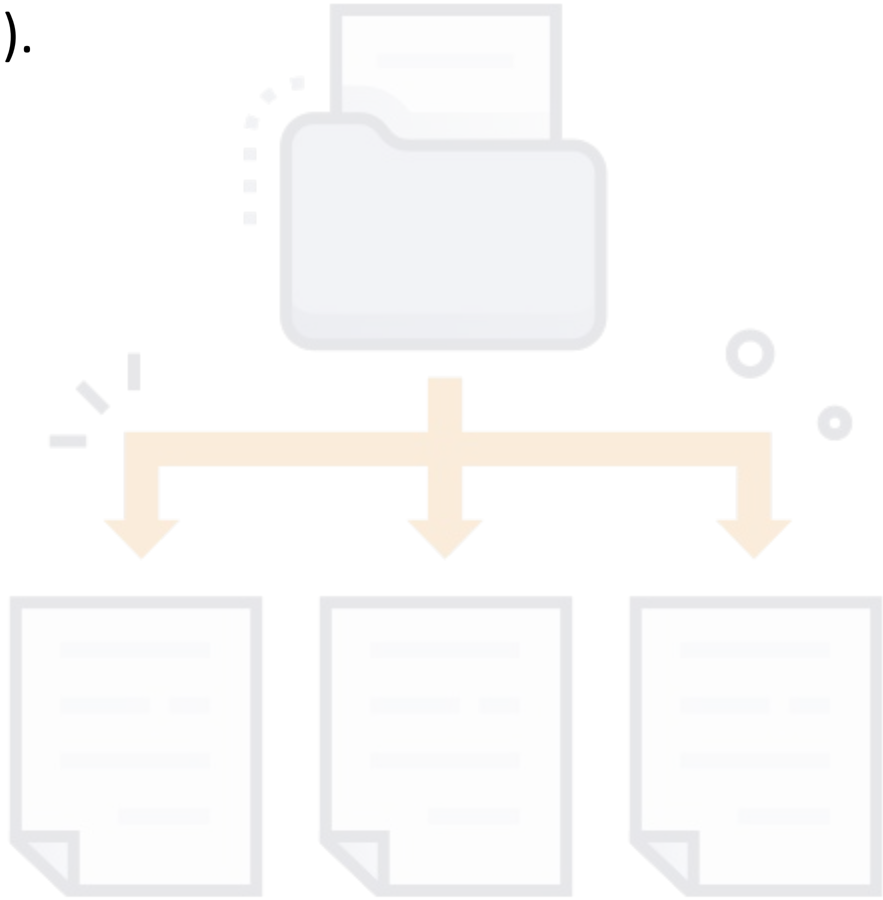
Key Notes:

- **Positive emotions:** "happy," "love," "good," "excited"
- **Everyday life:** "daily activities," "hobbies," "social interactions"
- **Gratitude and appreciation:** "thankful," "grateful," "proud"

Data Split

Data Preparation for Modeling

- Split data into features (title) and target (stress label).
- Split data into training (80%) and testing (20%) sets.



Data Overview

- **Key Findings:**
  - **Class imbalance:** More non-stressful articles than stressful ones.
  - **Overlapping text length distributions:** Text length alone may not be a strong predictor of stress.
  - **Distinct word patterns:** Different words are frequently used in stress and non-stress articles.

Deliver (1)

Feature Engineering

- **Key Features:**
  - Text data (primary feature)
  - Captures sentiment, emotions, and vocabulary
- **Business Significance:**
  - Text features are crucial for understanding the linguistic cues associated with stress.
  - Effective feature engineering can improve model performance and interpretability.
- **Techniques:**
  - **Text Cleaning:**
    - Removes irrelevant information (punctuation, stop words, hashtags).
  - **Text Normalization:**
    - Lowercases text for consistency.
  - **TF-IDF Vectorization:**
    - Converts text into numerical features representing word importance.

Deliver (2)

Machine Models Used

- **LinearSVC:** This is a linear support vector classifier used in the initial model. It is a good choice for text classification tasks due to its simplicity and efficiency.
- **LSTM (Long Short-Term Memory):** This is a recurrent neural network architecture used in the later model. LSTMs are well-suited for sequential data like text, as they can capture long-term dependencies between words.

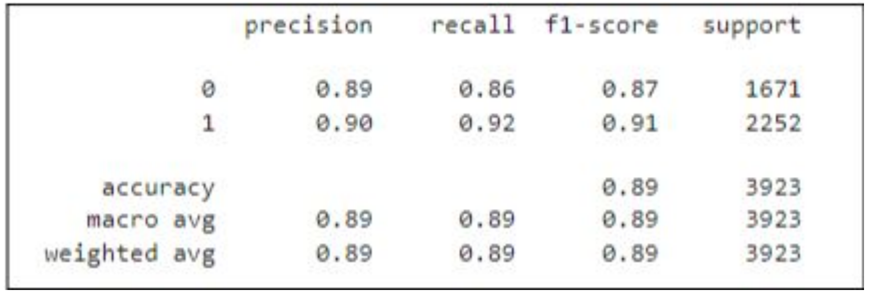
Deliver - LinearSVC (3)

Evaluation Metrics

- **Accuracy:** Measures the proportion of correct predictions made by the model.
- **Confusion Matrix:** Visualizes the number of correct and incorrect predictions for each class (stressful vs. non-stressful).
- **Classification Report:** Provides detailed information about the model's performance, including precision, recall, and F1-score for each class.
- **ROC AUC Score:** This metric is used for imbalanced datasets and measures the model's ability to distinguish between classes (AUC-ROC score closer to 1 indicates better performance).

Deliver - LinearSVC (5)

Evaluation Metrics - Confusion Matrix

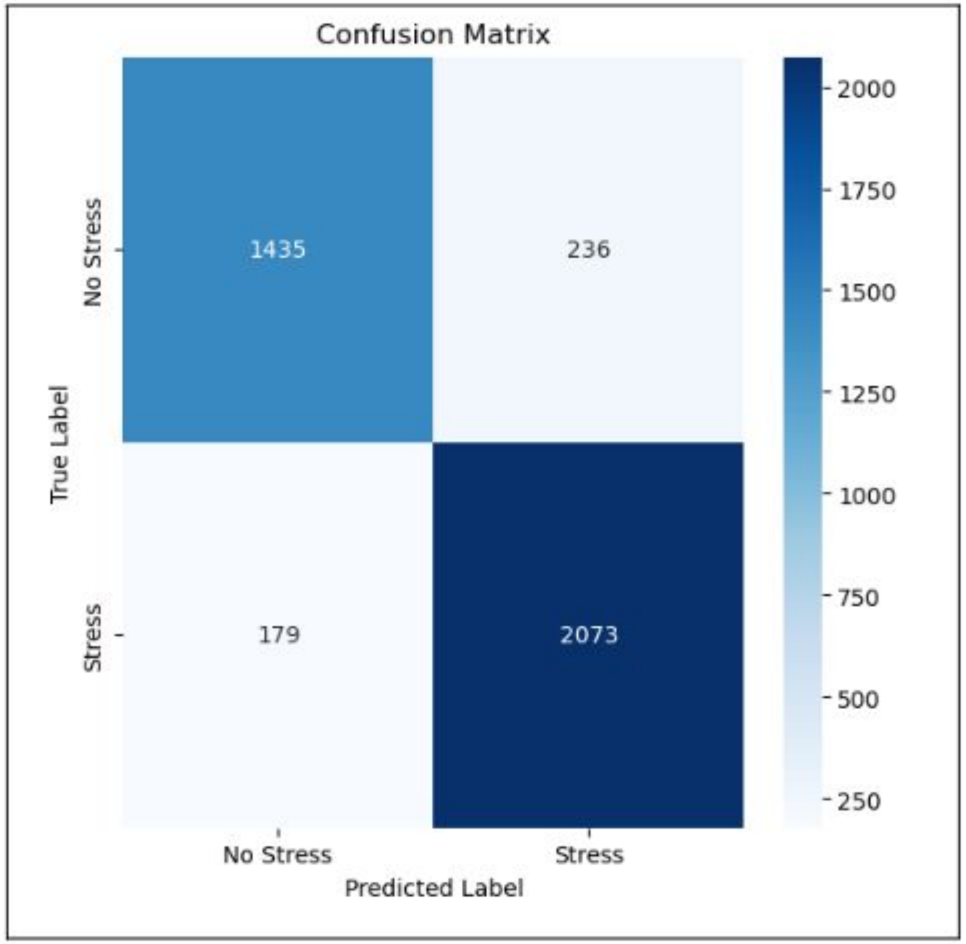


	precision	recall	f1-score	support
0	0.89	0.86	0.87	1671
1	0.90	0.92	0.91	2252
accuracy			0.89	3923
macro avg	0.89	0.89	0.89	3923
weighted avg	0.89	0.89	0.89	3923

- This confusion matrix visualizes the performance of the LinearSVC model in classifying stress and non-stress posts.
- **Key Findings:**
  - **Accuracy:** 89% of samples were correctly classified.
  - **Class-wise Performance:**
    - **Stress:** Precision: 0.90, Recall: 0.92, F1-score: 0.91
    - **Non-Stress:** Precision: 0.88, Recall: 0.85, F1-score: 0.87
- **Implications:**
  - **Overall good performance:** 89% accuracy.
  - **Class imbalance impact:** Model might be biased towards predicting non-stressful posts.

Deliver - LinearSVC (6)

Evaluation Metrics - Classification Report



This classification report summarizes the performance of the LinearSVC model on the test dataset.

Key Findings:

- **Overall Accuracy:** 89%
- **Class-wise Performance:**
  - **Stress:** Precision: 0.90, Recall: 0.92, F1-score: 0.91
  - **Non-Stress:** Precision: 0.88, Recall: 0.85, F1-score: 0.87
  - **Implications:**
- Strong overall performance.
- Potential impact of class imbalance.

Deliver - LSTM Model Training (7)

LSTM Model Training Progress

```
Epoch 1/10
442/442 ━━━━━━━━━━━━━━━━━━━━ 11s 22ms/step - accuracy: 0.7707 - loss: 0.4693 - val_accuracy: 0.8675 - val_loss: 0.3341
Epoch 2/10
442/442 ━━━━━━━━━━━━━━━━━━━━ 9s 21ms/step - accuracy: 0.9062 - loss: 0.2452 - val_accuracy: 0.8720 - val_loss: 0.3301
Epoch 3/10
442/442 ━━━━━━━━━━━━━━━━━━━━ 9s 20ms/step - accuracy: 0.9411 - loss: 0.1673 - val_accuracy: 0.8764 - val_loss: 0.3346
Epoch 4/10
442/442 ━━━━━━━━━━━━━━━━━━━━ 9s 21ms/step - accuracy: 0.9562 - loss: 0.1304 - val_accuracy: 0.8739 - val_loss: 0.3582
Epoch 5/10
442/442 ━━━━━━━━━━━━━━━━━━━━ 9s 21ms/step - accuracy: 0.9650 - loss: 0.1055 - val_accuracy: 0.8790 - val_loss: 0.3990
Epoch 6/10
442/442 ━━━━━━━━━━━━━━━━━━━━ 9s 21ms/step - accuracy: 0.9803 - loss: 0.0678 - val_accuracy: 0.8694 - val_loss: 0.4369
Epoch 7/10
442/442 ━━━━━━━━━━━━━━━━━━━━ 9s 21ms/step - accuracy: 0.9804 - loss: 0.0629 - val_accuracy: 0.8745 - val_loss: 0.5123
Epoch 8/10
442/442 ━━━━━━━━━━━━━━━━━━━━ 9s 21ms/step - accuracy: 0.9839 - loss: 0.0474 - val_accuracy: 0.8790 - val_loss: 0.6108
Epoch 9/10
442/442 ━━━━━━━━━━━━━━━━━━━━ 9s 21ms/step - accuracy: 0.9869 - loss: 0.0450 - val_accuracy: 0.8726 - val_loss: 0.5503
Epoch 10/10
442/442 ━━━━━━━━━━━━━━━━━━━━ 9s 21ms/step - accuracy: 0.9879 - loss: 0.0395 - val_accuracy: 0.8567 - val_loss: 0.5660
123/123 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step
```

This graph illustrates the training progress of the LSTM model, showing the changes in accuracy and loss over the epochs.

Key Findings:

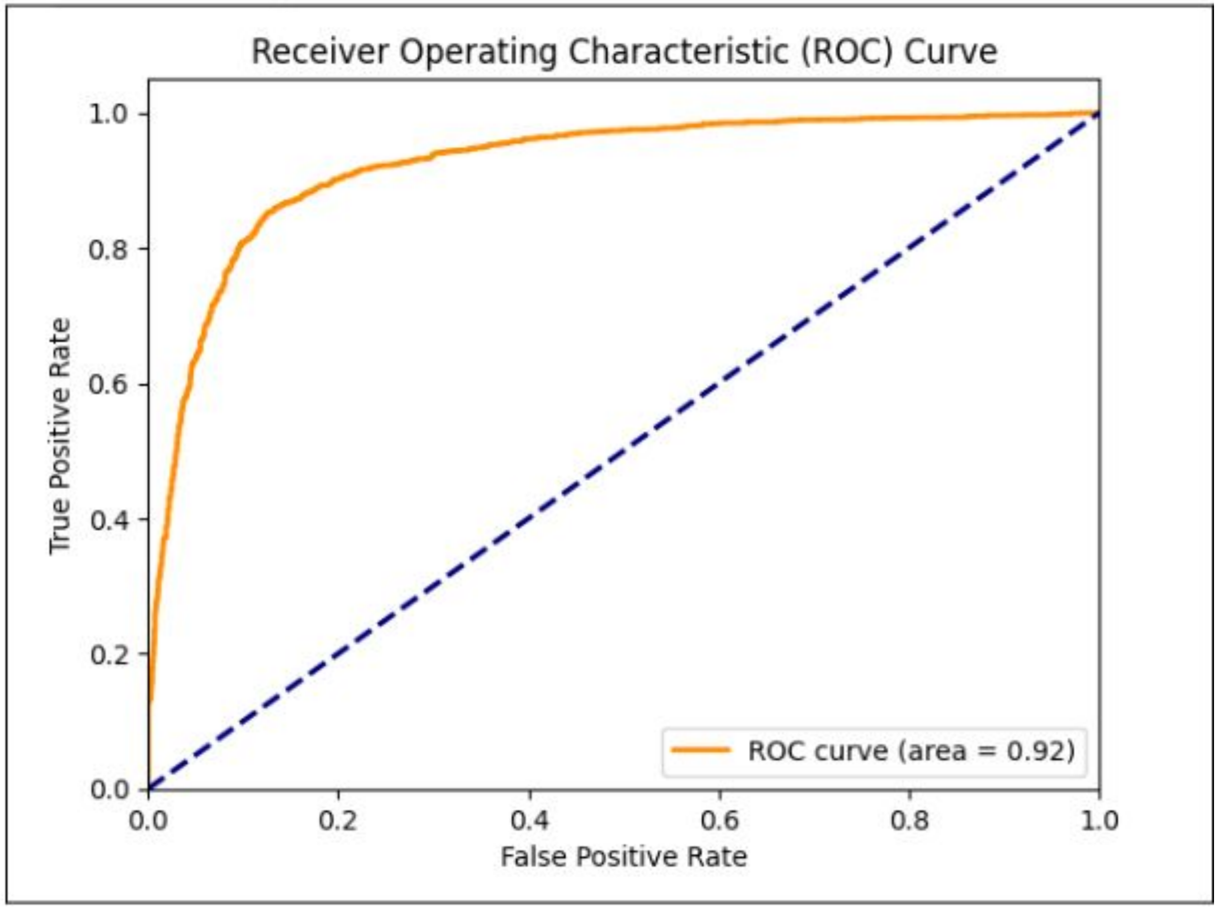
- **Improving Accuracy:** Model learned from data over time.
- **Decreasing Loss:** Model minimized errors.
- **Overfitting Potential:** Evidence of overfitting.

Implications:

- High training accuracy but potential overfitting.
- Explore early stopping or regularization.

Deliver - LSTM Model Training (7)

Evaluation Metrics - ROC AUC Score



This ROC curve illustrates the performance of the LSTM model in distinguishing between stress and non-stress posts.

Key Findings:

- **AUC-ROC Score:** 0.92, indicating strong overall performance.
- **Curve Shape:** Upward-sloping curve, suggesting good discrimination ability.

Implications:

- Effective model for stress detection.
- Threshold selection for desired balance between sensitivity and specificity.

Deliver (8)

Deployment - Stress Detection Web Application (Stress)

Steamlit:

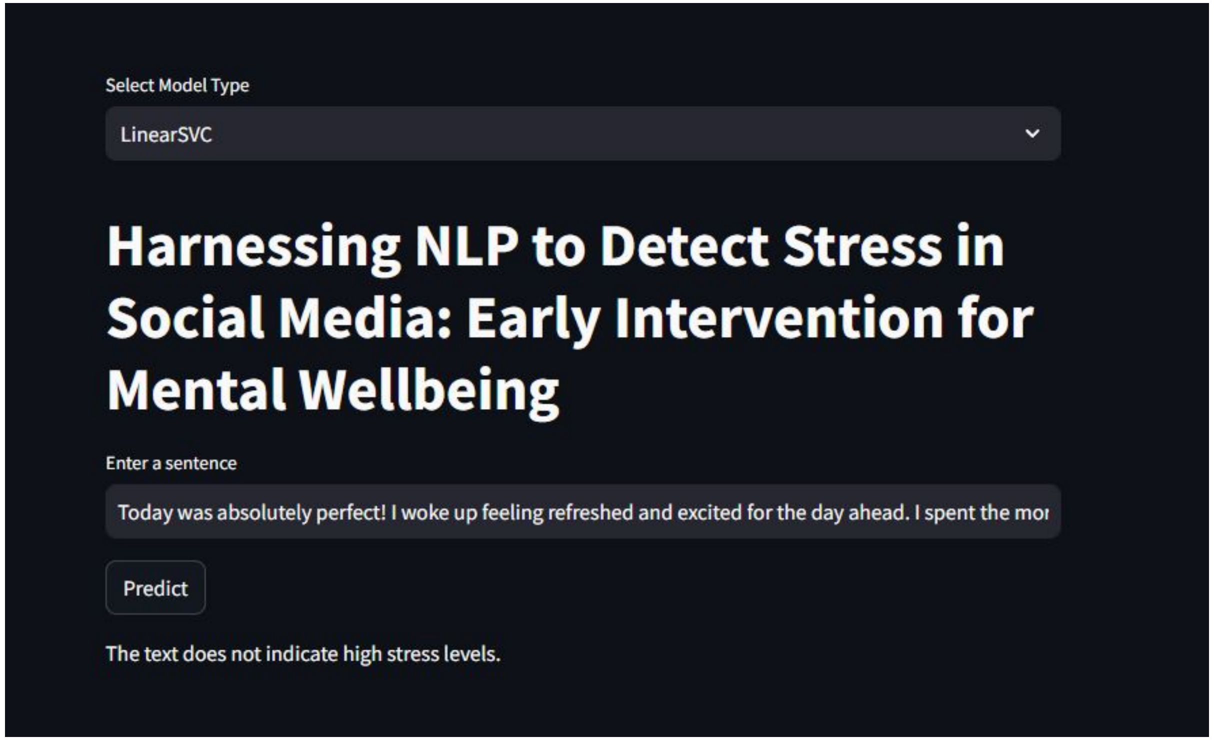
V1.0: https://socialmediastressdetection-7fzrqotzmpuwlwrcq82kpa.streamlit.app/

V0.2 (test): https://socialmediastressdetection-3rrnmlwq4coze4fgihuiq8.streamlit.app/

- **Sample sentence:** “I can’t believe I’m still up. It’s already midnight. I have so much to do tomorrow, and I feel like I haven’t accomplished anything today. My head is spinning, and I can’t seem to focus. I just want to crawl into bed and disappear. I’m so tired and overwhelmed. I don’t know how I’m going to make it through this week.”

Deliver (9)

Deployment - Stress Detection Web Application (Non - Stress)



Steamlit:

V1.0: https://socialmediastressdetection-7fzrqotzmpuwlwrcq82kpa.streamlit.app/

V0.2 (test): https://socialmediastressdetection-3rrnmlwq4coze4fgihuiq8.streamlit.app/

- **Sample sentence:** "Today was absolutely perfect! I woke up feeling refreshed and excited for the day ahead. I spent the morning enjoying a leisurely breakfast with my loved ones, followed by a long walk in the park. The sun was shining, the birds were singing, and I couldn't help but smile. I felt a sense of peace and contentment that I haven't experienced in a long time. It's truly amazing how a simple day filled with gratitude and joy can make such a difference."

Summary

- **Stress Detection Model:** Developed a successful NLP-based model to detect stress in social media text.
- **Model Performance:** Achieved high accuracy and F1-score, demonstrating strong performance in classifying stress and non-stress posts.
- **Class Imbalance:** Addressed the class imbalance issue through techniques like class weighting or oversampling.
- **Future Directions:** Considered potential improvements and future research directions.

Conclusions

- **Accurate Stress Detection:** Developed a model capable of accurately detecting stress in social media text.
- **Valuable Insights:** Gained insights into the linguistic patterns associated with stress and non-stress.
- **Potential for Early Intervention:** The model can be used for early identification of individuals at risk of stress.

Next Steps

- **Deployment:** Integrate the model into a real-world application (e.g., social media platform, mental health platform).
- **Data Collection:** Collect more diverse and representative data to improve model generalizability.
- **Model Refinement:** Explore advanced techniques like transfer learning or ensemble methods to further enhance performance.
- **Ethical Considerations:** Address ethical implications of using AI for mental health detection and ensure privacy and fairness.

Overall, this project demonstrates the potential of NLP-based approaches for early stress detection and highlights the importance of addressing ethical considerations in AI for mental health.

Appendices

References

Data Source:

https://www.kaggle.com/datasets/mexwell/stress-detection-from-social-media-articles

Source Code:

https://github.com/jimmychong1983/SocialMediaStressDetection

Steamlit:

V1.0: https://socialmediastressdetection-7fzrqotzmpuwlwrcq82kpa.streamlit.app/

V0.2 (test): https://socialmediastressdetection-3rrnmlwq4coze4fgihuiq8.streamlit.app/

Questions

Thank you
End of Presentation