MSML651: Big Data Analytics

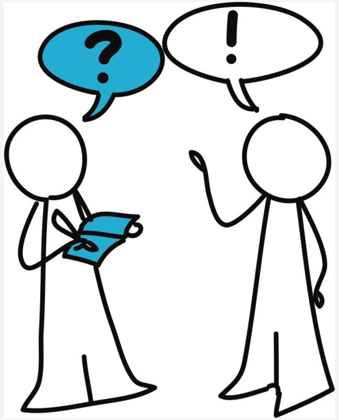
News Article Classification using Machine Learning

Presented By
Suraj T C

Problem Statement

The Core Problem:

- The dynamic nature of language and the expansive range of news topics make traditional methods of news classification less adaptable.
- As we grapple with this information explosion, there's a pressing need for advanced techniques to precisely categorize and understand the nuances within news articles.



The Dilemma:

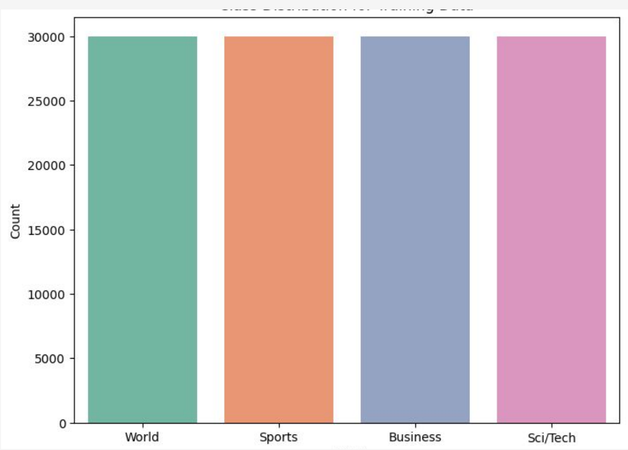
- Traditional approaches may struggle to keep pace with the evolving language and content of news articles.
- This creates a gap between the rapid evolution of news and the effectiveness of current classification systems.

Why It Matters:

- A proficient news classification system is crucial for efficient information retrieval, aiding researchers, analysts, and anyone seeking specific news topics.
- Solving this problem ensures that our systems can accurately interpret the vast landscape of news, making information more accessible and manageable.

Dataset Overview: AG News Classification

- The AG News dataset is a widely used **benchmark for text classification tasks**, specifically designed for news categorization.
- The dataset is sourced from the AG's corpus of news articles collected by the Academic Free License.
- Composition
  - Size: **127,600** News Articles
  - Categories: **Four** major categories, each representing a specific news genre.
    - 0 -> World
    - 1 -> Sports
    - 2 -> Business
    - 3 -> Science/Technology
- Data Distribution
  - Each category contains **30,000 train** samples and **1,900 test** samples



Data Distribution

Data Preprocessing with PySpark

Text Cleaning:

- Remove HTML Tags:
    - Eliminate any HTML tags from the text data.
- Remove Special Characters:
    - Discard non-alphanumeric characters, punctuation, and symbols.

Tokenization:

- Support Vector Classifier: Utilized Scikit-learn's **TfidfVectorizer()**.
- BERT: Leveraged transformers' **BertTokenizer()** for advanced tokenization.
- RoBERTa: Applied transformers' **RobertaTokenizer()** for robust tokenization.

Utilized Spark UDF for Efficient Data Processing

```
df_spark.sample(False, 0.01).limit(20).show()
```

```
+--------------------+-----+
|                text|label|
+--------------------+-----+
|Companies Approve...|    3|
|Microsoft Upgrade...|    3|
|Google: Now Playb...|    3|
|New Allergy Vacci...|    3|
|Venezuelan Presid...|    0|
|Swatch tax compla...|    2|
|Olympian on Brito...|    0|
|HP pushes parity ...|    3|
|Indonesia urges d...|    2|
|PivX hardens Wind...|    3|
|Iraq Conference i...|    0|
|Iraqi Delegation ...|    0|
|Oil Prices Slip B...|    2|
|Higher oil prices...|    0|
|Americans Miss Cu...|    1|
|Deaths in Nigeria...|    0|
|Stocks Up on Earn...|    2|
|Wireless vendors ...|    3|
|Friday the 13th P...|    3|
+--------------------+-----+
```

Pre Processed Data

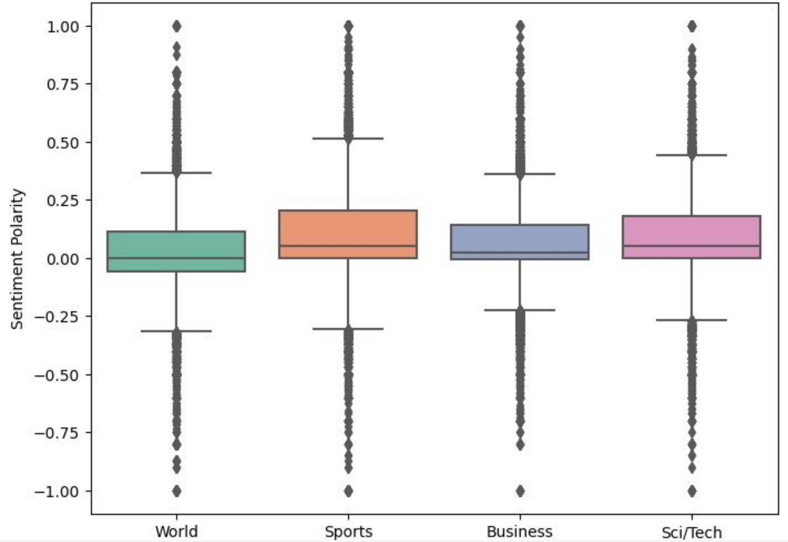
Data Exploration: Sentiment Analysis

Evaluate Bias through Sentiment Analysis:

- Employed **TextBlob** for sentiment analysis.
- Sentiment Analysis by Class
  - Visualized sentiment scores using a box plot.
  - Each class represented on the x-axis.
  - Sentiment polarity depicted on the y-axis.

Observation:

- Majority of data clustered around the **neutral** zone.
- Indicates a prevalence of neutral sentiment across all classes.



Sentiment Scores by Class

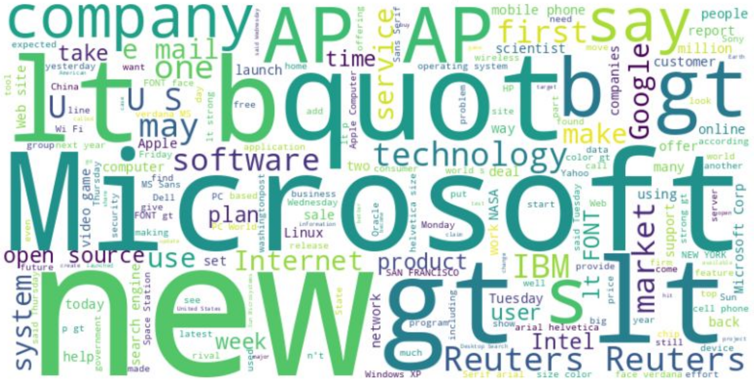
Data Exploration: World Cloud

Word Cloud for Class World

Word Cloud for Class Sports

Word Cloud for Class Business

Word Cloud for Class Sci/Tech



Model Training: SVC

Support Vector Classifier (SVC)

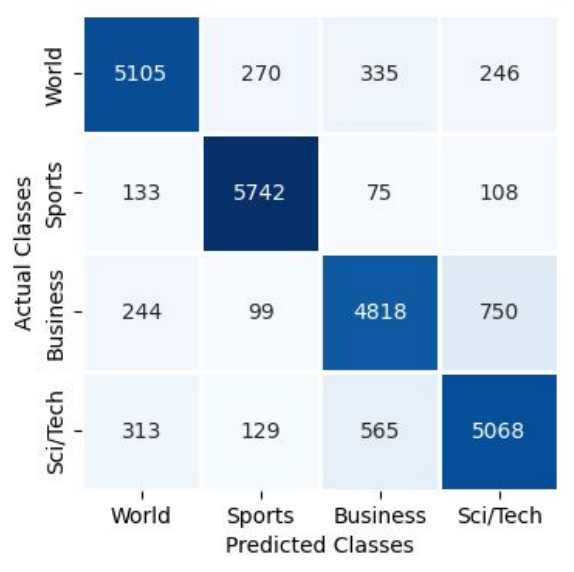
Challenge Encountered: Model execution on CPU caused crashes.

Data Reduction:

- Reduced training data to 24,000 samples per class.
- Tested the model on 6,000 samples per class for validation.

Algorithm:

- Sklearn Support Vector Classifier (SVC) with a **linear** kernel.
- Accuracy Achieved: **86.37%**.
- Training Time: **47 mins**.



Confusion Matrix

Model Training: BERT and RoBERTa

Fine-tuning BERT and RoBERTa:

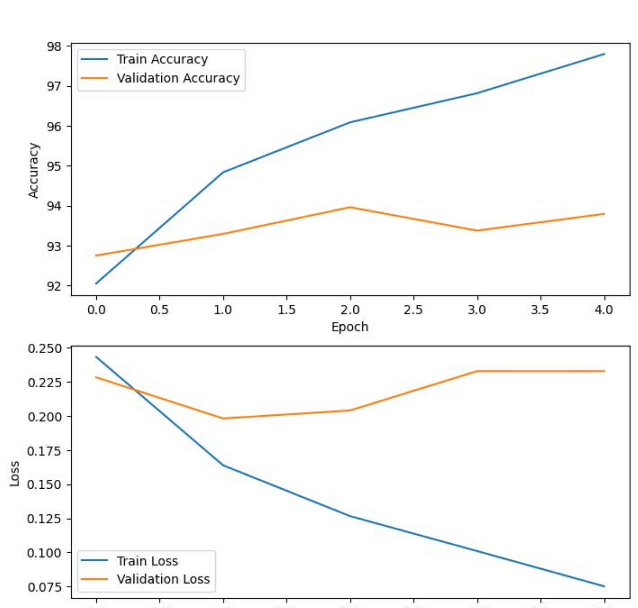
- Utilized Google Colab's **T4 GPU** for model training.
- Employed **pre-trained models** from the Hugging Face transformers.

Model Architecture:

- BERT:
  - Trained using tokenized data from BERT tokenizer.
  - Learning rate: **1e-4**, Epochs: 5, Batch size: 32.
- RoBERTa:
  - Trained using tokenized data from RoBERTa tokenizer.
  - Learning rate: **1e-5**, Epochs: 5, Batch size: 32.

Training Details:

- 3750 iterations per epoch for batch size of 32.
- Each epoch took an average of **90 mins**.
- Noted consistent validation metrics after **3 epochs**, indicating possible early stopping.



Train vs Validation Metrics

Result Evaluation

Model	Platform	Tokenizer	Batch Size	Training Time	Accuracy (%)	F1 Score
Logistic	CPU	N/A	N/A	12 mins	83.65	0.84
SVC	CPU	N/A	N/A	47 mins	86.38	0.85
BERT	T4 GPU	BERT	32	450 mins	88.46	0.88
RoBERTa	**T4 GPU**	**RoBERTa**	**32**	**450 mins**	**91.89**	**0.92**

Conclusion and Future Work

Conclusion

- In conclusion, the exploration and experimentation with different models for news classification have yielded insightful results
- Achieved impressive testing accuracies of 88.46% for BERT and 91.89% for RoBERTa
- RoBERTa is the best model among all three

Future Work

- Hyper-parameter tuning
- Topic modeling and multi-class classification
- Ensemble Models
- Bias Analysis

References

- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165.
- Hugging Face. (2021). Transformers Library Documentation. https://huggingface.co/transformers/
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.
- PySpark Documentation. (2021). https://spark.apache.org/docs/latest/
- TextBlob Documentation. (2021). https://textblob.readthedocs.io/en/dev/
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

THANK YOU

Any Questions?