Case Study

Roman Urdu Sentiment Identification

Senior Data Scientist Role - Amazon Pxt Group

Joseph Girsch

12/22/2023

Outline

1. Case Study Overview
2. Data Overview
3. Data Cleaning Summary
4. Language Detection
5. Feature Extraction
6. Model Development
7. Results
8. Feature Importances for Negative Sentiment Prediction
9. Data Limitations
10. Business Implications

Case Study Overview

- **Goal**: Train a sentiment classifier for the social media posts of the under-resourced *Roman Urdu* language.
  - Special focus on Negative sentiment over Positive or Neutral sentiments.

- **Challenges**:
  - Provided data is contaminated and requires processing.
  - Traditional NLP tools do not support *Roman Urdu*.

Data Overview

- The *Roman Urdu* data set is a limited corpus containing 20,229 records
- When processed, each record contains two columns:
  - A ‘Text’ column containing the corpus of social media posts
  - A ‘Sentiment’ column containing the general sentiment of the row (Positive, Neutral, or Negative)

Corpus sample

Text	Sentiment
Mare sayhat bachou k nak naseeb ke dua	Positive
Mari bati ur un ky ghar walon kyliay	Positive
Dua kar day Allah mujay zati rahish atta farmay.	Positive
jo log pehle se koi wazifa parh rahe hon	Neutral
Phali ky dar sy nahi aty	Negative

Sentiment	Row Count
Positive	6,013
Neutral	8,929
Negative	5,287

Data Cleaning Summary

Observation	Action Taken
No field headers	Designate field names
Corrupted Sentiment values	Regex to streamline
112 text rows containing only blank values	Rows deleted
All blank rows have a ‘Neutral’ sentiment	Nothing. Blank answers are valid responses and imply ‘Neutral’ sentiment
The 3rd column is 99.96% null	Dropped the column
Rows with other languages	**Intermediate language detection step required.**

Remaining rows after processing and removing duplicates: 19,648

Language Detection

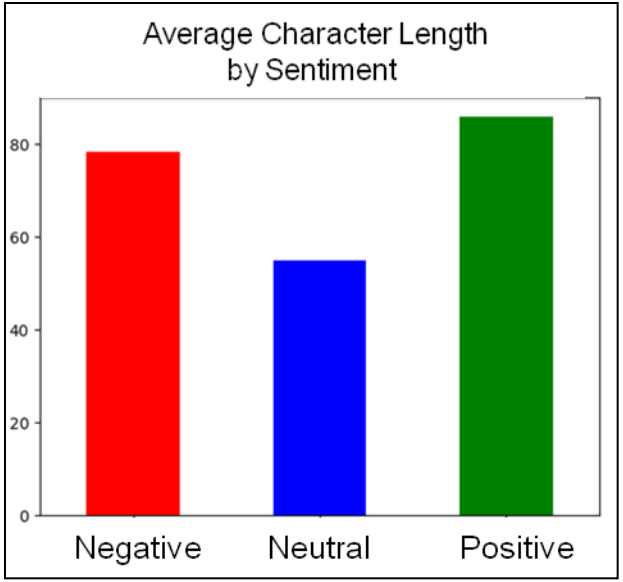
- **Challenge:** While language detection is supported for the Urdu language itself by Python's 'langdetect' and the AWS' Comprehend service, no commonly known package or service supports the *Roman script* of the Urdu language.

- **Solution:** Wrap batches of the corpus in prompt text and systematically loop through OpenAI's API to determine each row's dominant language.
  - Roman Urdu: 18,870*
  - Other languages removed: 778

*Results varies slightly from run to run

Feature Extraction

- Primary source of features: TF-IDF Vectorizer.
  - A TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer converts the *Roman Urdu* text into numerical features that machine learning algorithms can work with effectively.
  - See notes for more detailed explanation of TF-IDF Vectorizer.
- Text length was discovered to be a differentiating feature between Neutral sentiments and Positive or Negative sentiments, as seen in chart on right.



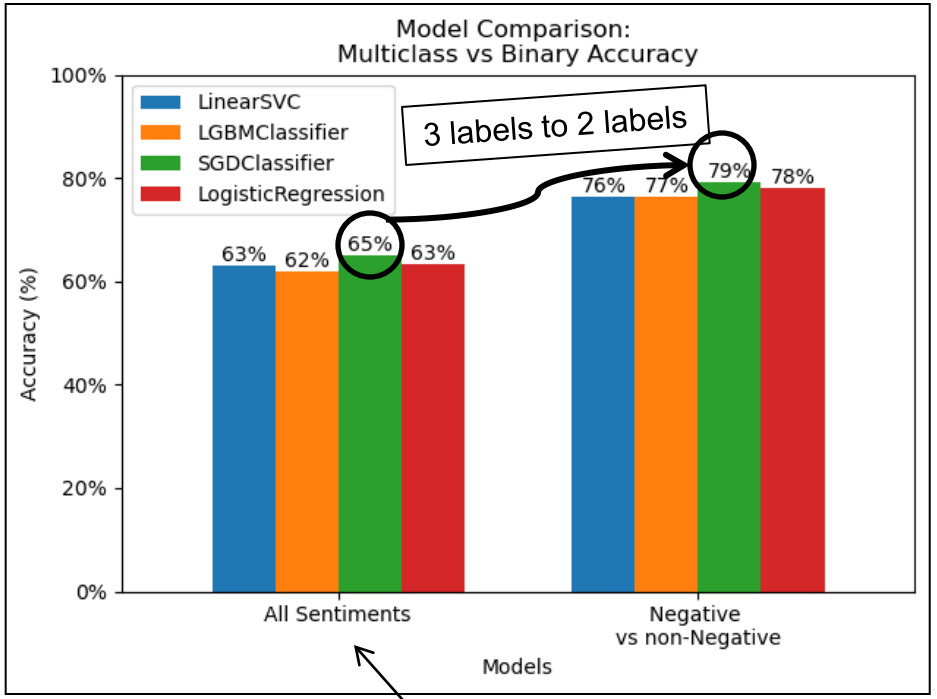
Model Development

- A sample of four models were evaluated for this case study.
  - Models were evaluated on an 80/20 training-test split using the micro-accuracy metric.
  - Model hyperparameters were tuned using the grid search cross validation technique.
- Metrics shown below.

Model	Micro-accuracy	"Negative" Accuracy	Weighted F1-Score	"Negative" F1-Score
LinearSVC	0.630	0.763	0.629	0.588
LGBMClassifier	0.621	0.765	0.617	0.541
SGDClassifier	0.649	0.792	0.647	0.589
LogisticRegression	0.634	0.782	0.632	0.584

Results

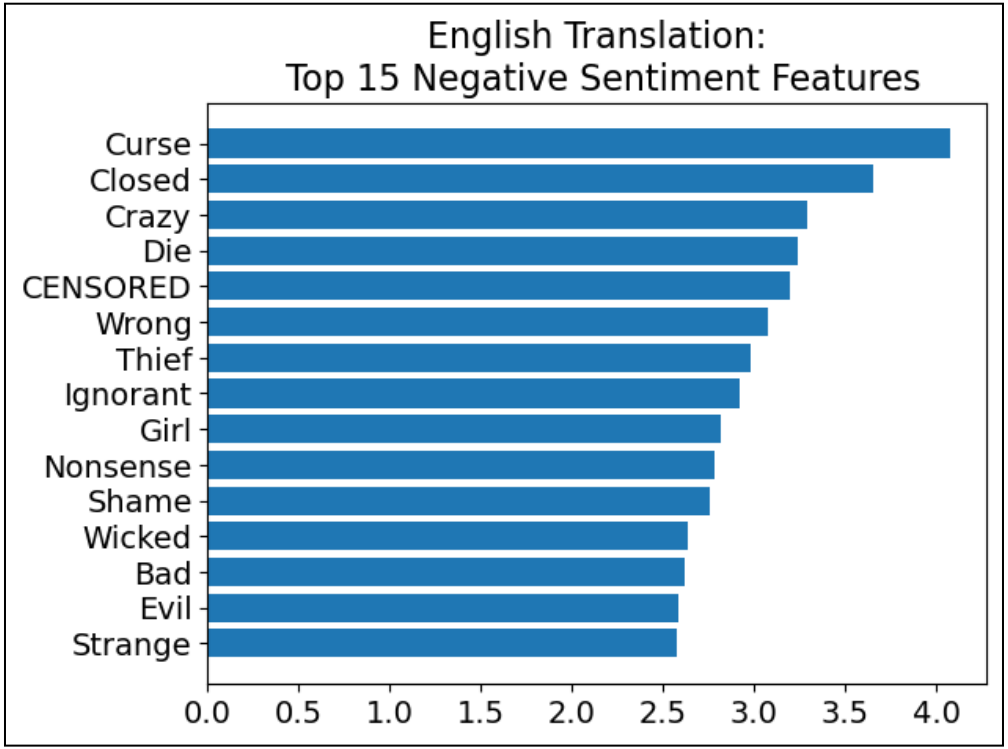
- The SGDClassifier had the best performance of all the models (64.9% Accuracy).
- However, if the stakeholders were willing to accept distinguishing only between **Negative** and **non-Negative** sentiments, Accuracy could be increased from 64.9% to **79.2%.**



Little difference in real performance between the four models.

Feature Importances for Negative Sentiment Prediction

- Displayed on the right are the strongest predictors of Negative *Roman Urdu* Sentiment.
  - Translation courtesy of the OpenAI API.
- As one can see, Social Media can be a scary place in any language.



Data Limitations

- **Limitation**:
  - What amounts to about 20,000 rows of corrupted sentence fragments is not a large body of text to train a language model in the grand scheme of machine learning.

- **Potential Solutions for Future**:
  - Perhaps a larger corpus of *Roman Urdu* text could be scraped from the web to improve sentiment prediction accuracy.

Summary

- Roman Urdu, while not currently well supported by language services, can be successfully modeled when used with AI-powered large language models.
- All machine learning models tested perform roughly on par with each other.
- More data is required to greatly increase prediction accuracy
- A binary Negative Sentiment detection model has higher accuracy for the same set of data than its multiclass counterpart.

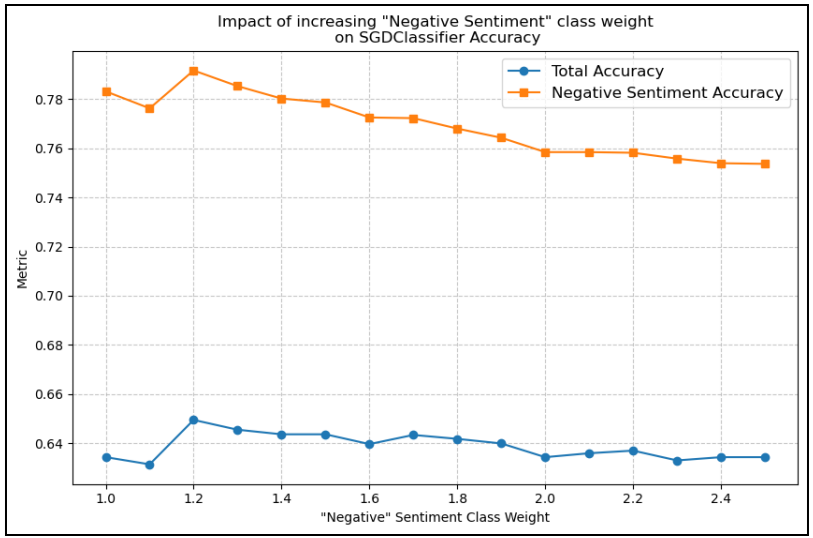
Business Implications

1. OpenAI API can act as a substitute for the ‘langdetect’ Python package or AWS’ Comprehend service until they support *Roman Urdu* language detection.
2. The simplified binary Negative Sentiment detection model requires less data to be accurate.

Backup

Lessons Learned

- *Urdu* and *Roman Urdu* are distinct languages.
  - I built and debugged a language detection model using boto3 only to realize that *Roman Urdu* wasn't supported.
- The chart on the right shows that no tradeoff exists between total accuracy and negative sentiment accuracy when adjusting class_weight toward the Negative Sentiment.
  - In fact adjusting the class_weight even gave a slight boost to both Accuracy measures before both began to degrade after Negative Sentiment exceeded a class_weight value of 1.2.



Python Packages Used

- Sklearn
- lightGBM
- OpenAI
- Matplotlib
- Re
- Pandas
- Numpy