Named Entity Recognition (NER) and Feature Engineering for Predictive Modeling

Objective

The goal of this task was to analyze news articles through Named Entity Recognition (NER) and engineer numerical features based on these entities for predictive modeling. This process aimed to predict article popularity using engagement metrics while exploring the relationship between named entities and engagement

Methodology

1. Data Collection

- Dataset: The dataset of news articles was obtained from Kaggle.
- Description: The dataset contains URLs to news articles, titles, and other metadata.
- Extraction:Articles' text content was extracted from URLs using requests and BeautifulSoup.A custom function extract_text_from_url handled HTML parsing and error management.
- Output: Extracted article text was saved as new_merged_data.csv.

2. Text Preprocessing

- Preprocessing ensured the text was clean and consistent for analysis:
- Removed HTML tags, special characters, and excessive whitespace.
- Converted all text to lowercase for normalization.
- Combined title and extracted_text into a single content column.
- Libraries Used: pandas, BeautifulSoup

3. Named Entity Recognition (NER)

- NER was performed using SpaCy's small English model (en_core_web_sm)
- Entities Extracted: PERSON, ORG (organizations), MONEY, GPE (geopolitical entities), and a custom CELEBRITY category.
- Custom Entity Rules:
  Recognized celebrities from a predefined list.
  Integrated this logic into SpaCy's pipeline.
- Outputs:
  Separate columns for entity types and counts (e.g. person_entities).
  Count of each entity type for feature engineering.

- Challenges Addressed:
  Handled long text inputs by limiting processing to manageable lengths.
  Libraries Used: SpaCy

4. Feature Engineering

- Features were created to capture the content's characteristics and engagement potential:
- Entity-Based Features:
  Counts of different entity types (person_count, org_count, etc.).
- Sentimental Features:
  Sentiment scores (TextBlob for polarity; VADER for sentiment).
- Engagement Features:
  popularity_score:found by ratio of tweet_count to the mean no of tweet
- Innovation: The inclusion of a custom CELEBRITY entity and its correlation with popularity.
- Libraries Used: TextBlob, VADER

5. Predictive Modeling

- A model was built to predict popularity_score using the engineered features:
- Model Selection:
  DecisionTreeRegressor
  GradientBoostingRegressor
  XGBoostRegressor
- Grid Search: Performed hyperparameter tuning using GridSearchCV with 5-fold cross-validation.
- Evaluation Metrics:
  MAE (Mean Absolute Error)
  MSE (Mean Squared Error)
- Best Model: GradientBoostingRegressor achieved the lowest MAE and MSE on the test set in Grid Search CV method.

6. Visualization

- To analyze relationships between features and popularity:
- Correlation Heatmap:
  Correlations between popularity_score and entity-based features.
- Actual vs Predicted Values:
  Line plot comparing y_test and y_pred for model performance.

Findings and Insights

Named Entities and Popularity:

- Articles featuring a higher count of ORG entities showed a slight positive correlation with popularity.
- PERSON entities had a weaker relationship, possibly dependent on article context.
- The custom CELEBRITY entity type showed no significant direct correlation with popularity.

Sentiment and Engagement:

- Articles with higher positive sentiment scores (VADER) tended to have higher popularity scores.
- TextBlob's polarity scores correlated less strongly with popularity than VADER's compound sentiment.

Model Performance:

- GradientBoostingRegressor outperformed DecisionTreeRegressor and XGBoostRegressor:
  MAE: 0.0054
  MSE: 0.0004

Conclusion

The given tasks, including text normalization, feature engineering, model selection, and correlation analysis, were executed effectively. The analysis reveals that the classification of news as fake or real has no significant impact on its popularity. Instead, sentiment scores and entity features, such as organizations, locations, and people, are the primary drivers of engagement. The GradientBoostingRegressor model, optimized via GridSearchCV, effectively captured these relationships, making it the best predictive model. These findings highlight the importance of emotionally engaging content and relevant entities in driving article popularity.