Detecting Fake News: A Machine Learning Approach

Unveiling the Role of AI in Curbing Misinformation

JOHN ANUGRAH PETER 23215206
SHRESHTH SHARMA 23215219
KARAN KUMAR 23215224
SUMIT NAGESIA 23215214

Understanding Fake News: A Growing Challenge

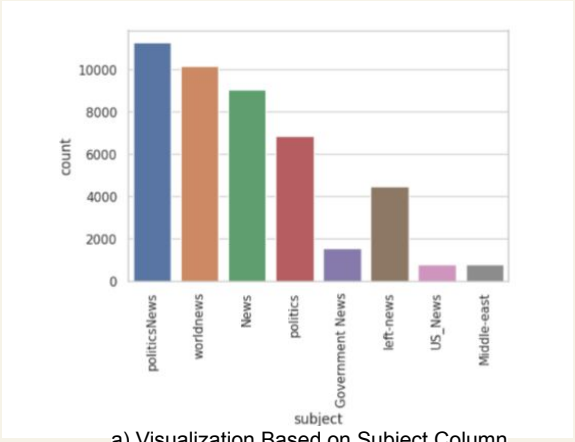
The uncontrolled spread of false information in the digital era threatens societal harmony and trust. This study leverages machine learning algorithms and Python to address this critical issue through advanced fake news detection. By exploring and evaluating cutting-edge models, it aims to:

- Strengthen decision-making.
- Uphold information integrity.
- Mitigate the adverse impacts of misinformation.

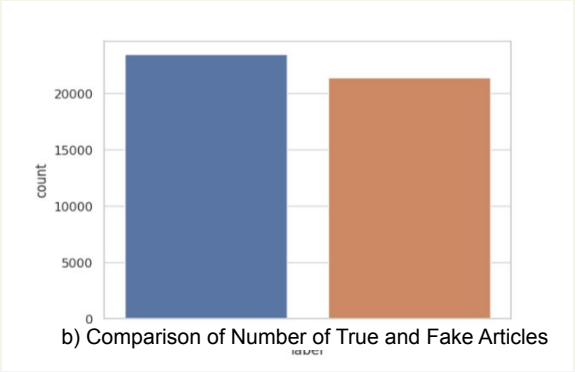
The study's implications extend to bolstering trust in media, safeguarding democratic processes, and ensuring content authenticity. Its findings benefit stakeholders, including news organizations, social media platforms, and governmental institutions.

Dataset Selection and Exploratory Data Analysis (EDA)

- **Dataset Used:** ISOT Fake News Dataset
  - Features: Title, text, type, and publication date.
  - Balanced composition of articles from reliable and unreliable sources.
- **Exploratory Data Analysis (EDA):**
  - Visualizations (e.g., bar graphs) explored subject distribution and key topics.
  - Insights revealed potential features to distinguish between true and fake news.
  - Verified no data imbalance for unbiased model training.
- **Preprocessing:**
  - Removed numbers, punctuation, and stopwords.
  - Improved text quality for higher model accuracy.



a) Visualization Based on Subject Column



b) Comparison of Number of True and Fake Articles

Metric and Model Selection

1. **Accuracy:**
   - Measures overall correctness of predictions.
   - Indicates the model's reliability in distinguishing between fake and real news.

$$\text{Formula: Accuracy} = \frac{TP+TN}{\text{Total Samples}}.$$

2. **Precision:**
   - Focuses on correctly identifying true positives (real news).
   - Critical to avoid the spread of fake information by minimizing false positives.

$$\text{Formula: Precision} = \frac{TP}{TP+FP}.$$

3. **Recall:**
   - Assesses the model's ability to identify real news without misclassifying it as fake.
   - Ensures the credibility of legitimate news sources is maintained.

$$\text{Formula: Recall} = \frac{TP}{TP+FN}.$$

News	Size (Number of articles)	*Subjects*	
Real-News	21417	**Type**	**Articles size**
		World-News	*10145*
		Politics-News	*11272*
Fake-News	23481	**Type**	**Articles size**
		Government-News	*1570*
		Middle-east	*778*
		US News	*783*
		left-news	*4459*
		politics	*6841*
		News	*9050*

Figure 2: Number of Articles Per Category

1. **F1-Score:**
    - Balances precision and recall, useful when both errors are equally important.
    - Provides a comprehensive evaluation of the model.
2. **AUC-ROC Score:**
    - Measures the model's ability to rank true positives higher than false positives.
    - Indicates how well the model differentiates between fake and real news.
3. **Confusion Matrix:**
    - Visual representation of model performance.
    - Shows counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Models Explored and Feature Encoding

Models Explored

- **Logistic Regression:**
  - Baseline model for binary classification.
  - Chosen for simplicity and interpretability.
  - **Optimized with:** Intel Extension for Scikit-learn to reduce runtime.
- **Advanced Models:**
  - Decision Tree, Random Forest, Gradient Boosting, XGBoost, Passive Aggressive Classifier.
  - Key Features:
    - Capture non-linear relationships and complex interactions.
    - Leverage ensemble learning and regularization for better accuracy.

Feature Encoding: TF-IDF Vectorization

- Converts text data into numerical representations.
- Weighs terms based on their importance in individual documents and across the dataset.
- Reduces noise and enhances model performance.

Why These Models?

- **Logistic Regression:** Serves as a reliable benchmark for comparison.
- **Gradient Boosting & XGBoost:**
    - Excel in handling complex datasets with ensemble learning.
    - Regularization and non-linear relationship handling boost performance.

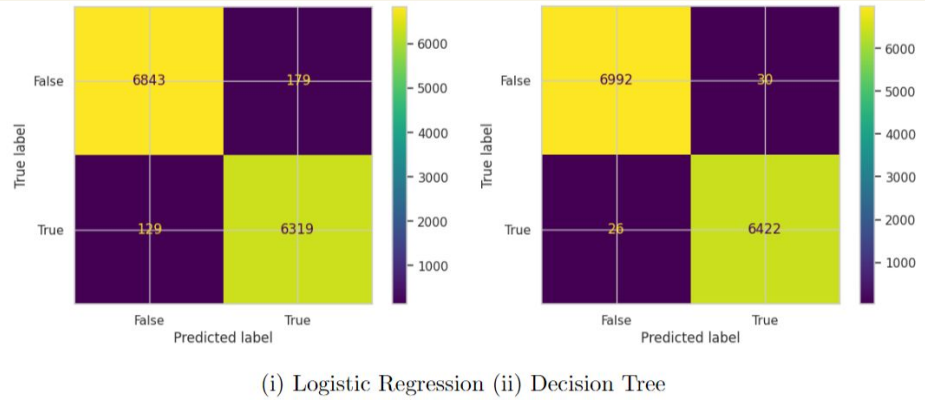
Model Evaluation and Results

Key Observations

- **Unexpected Outcome:**
    - Decision Tree, Gradient Boosting, and XGBoost outperformed the expected Passive Aggressive Classifier.

Reasons for Superior Performance:

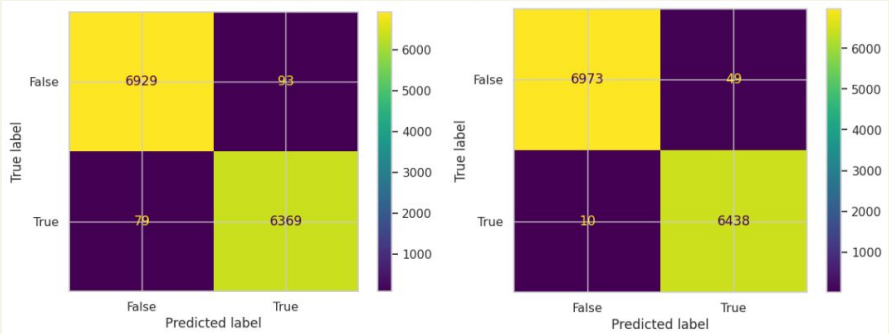
- Ability to handle **non-linear relationships.**
- Utilization of **ensemble learning** and **regularization techniques.**
- Adaptation to **dataset characteristics.**



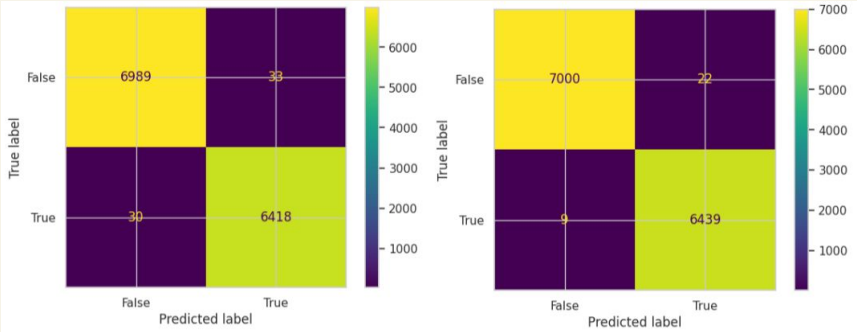
(i) Logistic Regression (ii) Decision Tree

XGBoost: The Best Performer

- **Insights from Confusion Matrices:**
  - Lowest **False Positives** and **False Negatives.**
  - Achieved the **highest training and testing accuracy:**
    - Minimal overfitting.
    - Excellent generalization capability.
- **Metric Performance:**
  - Top scores in **Precision, Recall, F1 Score, and ROC AUC.**
  - Demonstrated exceptional class separability.



(iii) Random Forest (iv) Gradient Boosting



(v) Passive Aggressive Classifier (vi) XGBoost

Conclusion:

- XGBoost emerged as the optimal model for this dataset, combining:
  - High accuracy.
  - Robust performance.
  - Balanced metrics for reliability.

Machine Learning Algorithm	Training Accuracy	Testing Accuracy	Testing Precision	Testing Recall	Testing F1-Score	ROC AUC Score
Logistic Regression	0.979954	0.977134	0.972453	0.979994	0.976209	0.977251
Decision Tree	1.0	0.995843	0.995350	0.995968	0.995659	0.995848
Random Forest	1.0	0.987231	0.985608	0.987748	0.986677	0.987252
Gradient Boosting	0.996786	0.995620	0.992446	0.998449	0.995439	0.995736
Passive Aggressive Classifier	1.0	0.995323	0.994885	0.995347	0.995116	0.995324
XGBoost	1.0	0.997699	0.996595	0.998604	0.997599	0.997736

Figure 5: Algorithm Performance Metrics

Conclusion and Future Works

Conclusion

- **XGBoost:**
  - Emerged as the most accurate machine learning model.
  - Demonstrated excellent performance with high precision, recall, and generalization.
- **Intel Optimized Logistic Regression:**
  - Reduced code runtime by **1.8x**, showcasing the value of optimization tools.

Future research will focus on enhancing data processing through techniques like stemming and lemmatization. It will explore algorithms such as Naïve Bayes, SVM, KNN, and MLP, while experimenting with advanced vectorizers like CountVectorizer, Word2Vec, BERT, and Gensim. Parameter optimization will be tackled using Grid Search and Random Search. Additionally, the research will extend to various real-world datasets and machine learning challenges, incorporating Python-based projects with advanced algorithms.