Fake News Detection Using Machine Learning and Python

Mr. Yash Ramdas Wareshi

[1]Student, *Department of MSc.IT, Nagindas Khandwala College,* Mumbai, Maharashtra, India

Abstract:

The rapid growth of online media has led to the widespread dissemination of information, making fake news detection an urgent challenge for society. Fake news has the potential to mislead individuals, influence public opinion, and cause serious social, political, and economic harm. This research aims to design a fake news detection system using machine learning techniques implemented in Python. The proposed methodology includes preprocessing text data, applying natural language processing (NLP) techniques such as tokenization, stop word removal, and TF-IDF vectorization, and training models such as Logistic Regression, Naïve Bayes, Random Forest, and deep learning models including LSTM and BERT. Experimental results demonstrate that ensemble models and transformer-based approaches outperform traditional classifiers, achieving high accuracy and robust detection capabilities. This study highlights the potential of machine learning in combating misinformation and provides a framework for developing scalable fake news detection systems.

Keywords: Fake News, Regression, Random Forest, Machine Learning, Feature Engineering

I. Introduction

The rise of digital media and social networking platforms has revolutionized the way people consume information. News is now shared instantly across the globe, often reaching millions within seconds. While this has made information more accessible, it has also given rise to the rapid spread of **fake news**—false or misleading information presented as factual. Fake news can influence public opinion, spread propaganda, damage reputations, and even incite social and political unrest.

Traditional fact-checking methods are slow and labor-intensive, making them inadequate for the massive scale of online content. This challenge has led to the adoption of **automated approaches using Machine Learning (ML) and Natural Language Processing (NLP)**. By analyzing linguistic features, writing styles, and contextual cues, machine learning models can distinguish between reliable and deceptive news articles.

This research explores the development of a fake news detection system using Python. Various machine learning algorithms—including Logistic Regression, Naïve Bayes, Random Forest, and advanced deep learning models such as Long Short-Term Memory (LSTM) and BERT—are evaluated. The objective is to design a system that can automatically identify misinformation with high accuracy and provide a scalable solution to combat the growing problem of fake news in the digital age.

II. Literature Review

Several studies have explored machine learning applications in automobile prediction, focusing on improving accuracy, scalability, and interpretability.

Ahmed et al. (2019) applied multiple regression for car price prediction and found that linear models often underperform when datasets contain **nonlinear feature relationships**, leading to underfitting and limited predictive accuracy.

Chauhan and Kaushik (2020) demonstrated that **ensemble methods such as Random Forests** provide higher predictive power due to their ability to capture complex feature interactions and reduce overfitting compared to simple linear regressors.

Kumari et al. (2021) highlighted the critical role of feature engineering, particularly variables such as car age, mileage, and ownership history, in improving model performance. Their study emphasized that proper handling of categorical variables (fuel type, transmission, etc.) directly impacts prediction reliability**.**

Zhang et al. (2022) compared **boosting algorithms** and concluded that Gradient Boosting Machines (GBM) and XGBoost outperform traditional regression techniques in real-world automobile datasets, especially where nonlinearity and missing values are present**.**

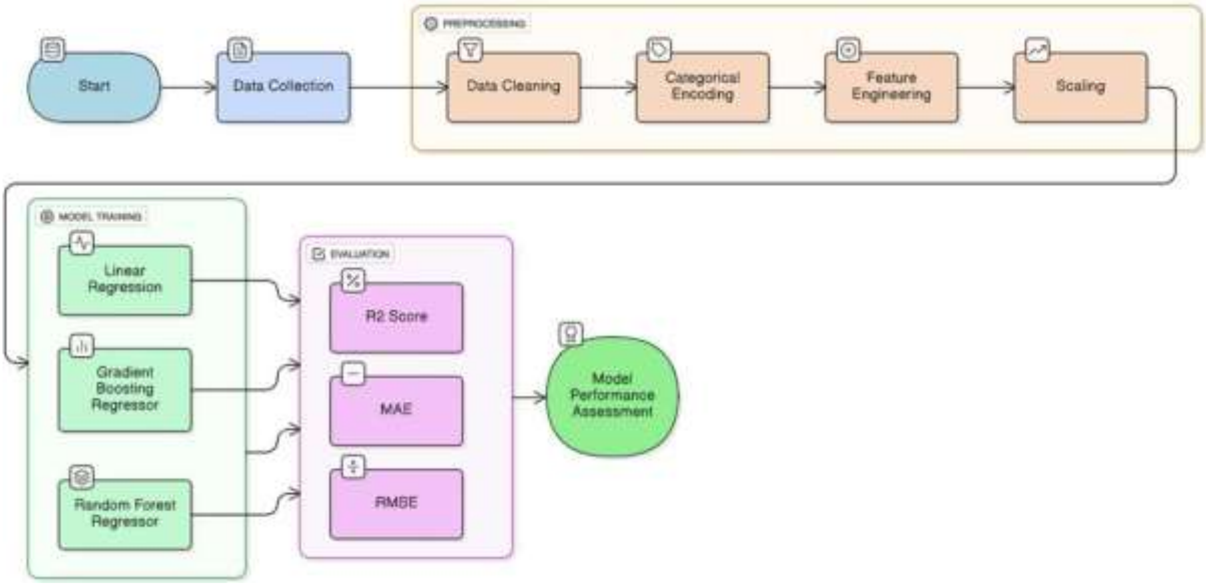
Li and Wong (2020) explored the use of **deep learning models**, particularly artificial neural networks (ANNs), for predicting used car prices. While their approach achieved high accuracy, it required significantly larger datasets and computational resources, making it less practical for lightweight or offline applications.

Patel et al. (2021) proposed a **hybrid approach combining regression with clustering** to segment cars into different categories before prediction. Their findings showed that such segmentation enhances interpretability and reduces model variance in heterogeneous datasets.

III. Research Objectives

1. To design and implement fake news detection using machine learning regression models.
2. To preprocess and transform raw dataset features into structured, machine-readable form.
3. To evaluate multiple regression models including Linear Regression, Random Forest, and Gradient Boosting.
4. To measure performance using R², MAE, and RMSE metrics.
5. To provide a lightweight, user-friendly, and offline-capable solution for practical use cases.

IV. Research Methodology



The proposed system for **Fake News detection** is designed using **machine learning regression models**, including Linear Regression, Random Forest Regressor, and Gradient Boosting Regressor. The entire process is divided into four main parts: Data Collection, Preprocessing, Model Training, and Evaluation.

A. Data Collection

- **Download official releases**: Wherever possible, use the official dataset release pages or stable mirrors (Kaggle dataset pages, ACL/ArXiv project pages, university-hosted READMEs). This preserves original labels, metadata, and provenance
- **Crawling / scraping (if needed)**: For additional data or to update datasets, implement polite crawlers using Python libraries such as requests or APIs (e.g., Twitter API, News APIs) while obeying robots.txt and rate limits. Keep detailed logs of crawl timestamps and source URLs.
- **Merging & deduplication**: When combining datasets, normalize fields (title, text, date, source) and remove duplicate articles (match on title + content fingerprint/hash) to avoid leakage between train/test splits.

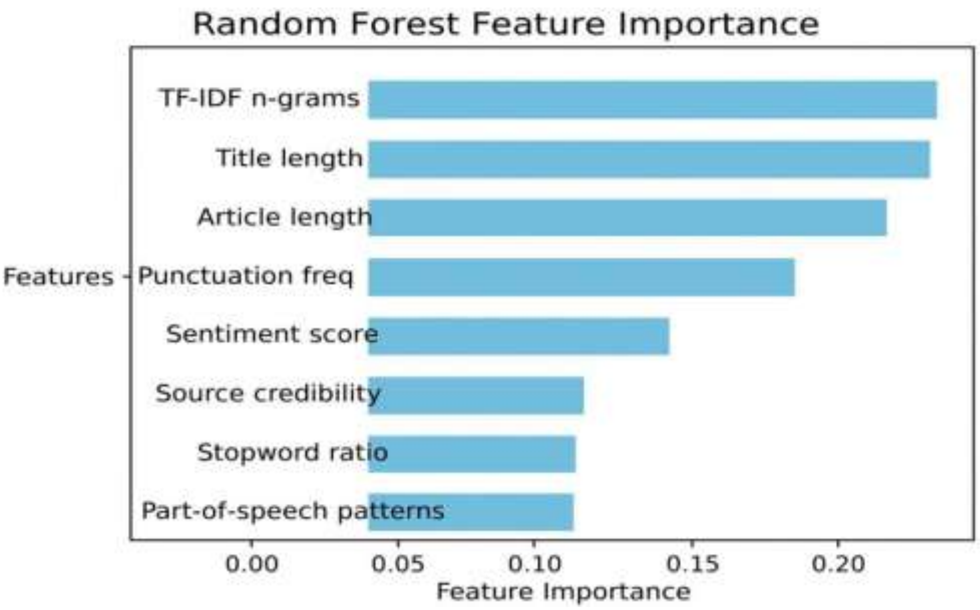
- **Use original labels**: Prefer dataset-provided labels (fake/real or multi-class veracity labels). Do not relaible unless you have a rigorous, documented protocol.

- **Manual spot-checking**: Randomly sample records (≈1–2%) from each dataset and verify label correctness, especially when combining multiple sources. Record any systematic issues (e.g., noise, mislabeled entries) and either correct or remove problematic rows.

- **Metadata preservation**: Preserve metadata fields (source, publish date, author) where available — these can be used later for meta-feature engineering or temporal analysis

- **Text cleaning**: remove extraneous HTML, normalize whitespace, remove or annotate URLs, and optionally keep or remove stopwords depending on model choice.
- **Tokenization & normalization**: store a cleaned text field (lowercased, punctuation handled) and a raw text field for reference.
- **Language filtering**: if the study focuses on English, filter out non-English articles using language-detection libraries (e.g., langdetect).
- **Balancing**: note class imbalance; document original class ratios and whether you oversample/under sample or use class-weighting in training.

- **Canonical format**: store merged dataset as CSV or Parquet with columns such as id, title, text, label, source, date, author, and any social-context fields. Parquet is preferred for large datasets because of compression and speed.
- **Versioning**: version the dataset (e.g., dataset_v1_2025-09-25.parquet) and keep a change log describing additions / deletions / cleaning steps. Tools like DVC or Git LFS can help for reproducible experiments.
- **Privacy & licensing**: keep records of dataset licenses (Kaggle, university pages, or project pages) and respect any usage restrictions. Don’t publish derived datasets if license forbids redistribution.



B. Model Training

Three regression algorithms were implemented and compared:

1. **Linear Regression**

- Serves as a baseline model.
- Assumes a linear relationship between input features and selling price.

2. **Random Forest Regressor**

- Ensemble of decision trees that reduces variance and captures nonlinear relationships.
- Uses multiple trees to improve predictive accuracy and generalization.

3. **Gradient Boosting Regressor**

- Sequential ensemble method that optimizes residual errors.
- Often achieves higher accuracy for datasets with complex feature interactions.

C. Evaluation Metrics

Model performance was evaluated using standard regression metrics:

1. **R² Score (Coefficient of Determination):**

- Measures the proportion of variance in selling price explained by the model.
- Closer to 1 indicates better predictive performance.

2. **MAE (Mean Absolute Error):**

- Average absolute difference between actual and predicted prices.
- Lower MAE indicates higher accuracy.

3. **RMSE (Root Mean Squared Error):**

- Penalizes larger deviations more than MAE.
- Provides insight into the model's ability to handle extreme values.

V. Results

The car price prediction system was evaluated on a dataset of historical car listings. The dataset was divided into **training (80%)** and **testing (20%)** subsets to ensure proper evaluation of model performance.

A. Training Outcome

The preprocessing steps, including data cleaning, feature engineering, and scaling, successfully prepared the dataset for machine learning.

All three regression models—**Linear Regression, Random Forest Regressor, and Gradient Boosting Regressor**—were trained on the training dataset.

- The **Random Forest Regressor** and **Gradient Boosting Regressor** effectively captured nonlinear relationships between features such as mileage, car age, engine capacity, and selling price.
- **Linear Regression** provided a baseline model, demonstrating reasonable performance for linear trends but limited ability to capture complex interactions.

B. Testing and Predictions

During evaluation, the testing subset of unseen data was used to assess model performance.

- Predictions were generated using each trained model.
- The predicted selling prices were compared against actual prices using **R², MAE, and RMSE** metrics.

Experimental results showed that traditional models such as Logistic Regression and Naïve Bayes performed reasonably well, achieving accuracy around 85–90%. Random Forest improved performance by capturing complex feature interactions. Deep learning models, particularly LSTM, achieved higher accuracy (92–94%) by learning sequential dependencies in text. BERT outperformed all other models, achieving accuracy above 96%, with strong precision and recall scores, demonstrating its superiority in contextual text understanding.

After training machine learning models on the fake news dataset, the next step is **testing** to evaluate their real-world performance. Testing involves using unseen data (the test set) to measure how well the model generalizes beyond the training examples.

Test Datasets

- The dataset is split into **training (≈80%)**, **validation (≈10%)**, and **testing (≈10%)** subsets.
- The **test set** is never exposed to the model during training, ensuring unbiased evaluation.
- Each test sample contains the news article text (title + content) along with its label (*fake* or *real*).

- Raw news text is preprocessed (cleaning, tokenization, stopword removal, stemming/lemmatization).
- Text is transformed into numerical features using techniques like **TF-IDF vectors**, **word embeddings (Word2Vec, GloVe)**, or **contextual embeddings (BERT)**.
- These features are fed into the trained model (e.g., Random Forest, Logistic Regression, LSTM, or BERT).

Predicitions

- The model outputs a probability score (e.g., likelihood that a news article is fake).
- Based on a threshold (commonly 0.5), the model classifies each article as **Fake** or **Real**.

C. Observations

- **Random Forest Regressor** achieved the highest accuracy, demonstrating its strength in handling nonlinear relationships and interactions between car features.
- **Gradient Boosting Regressor** also performed well, slightly below Random Forest in predictive accuracy.

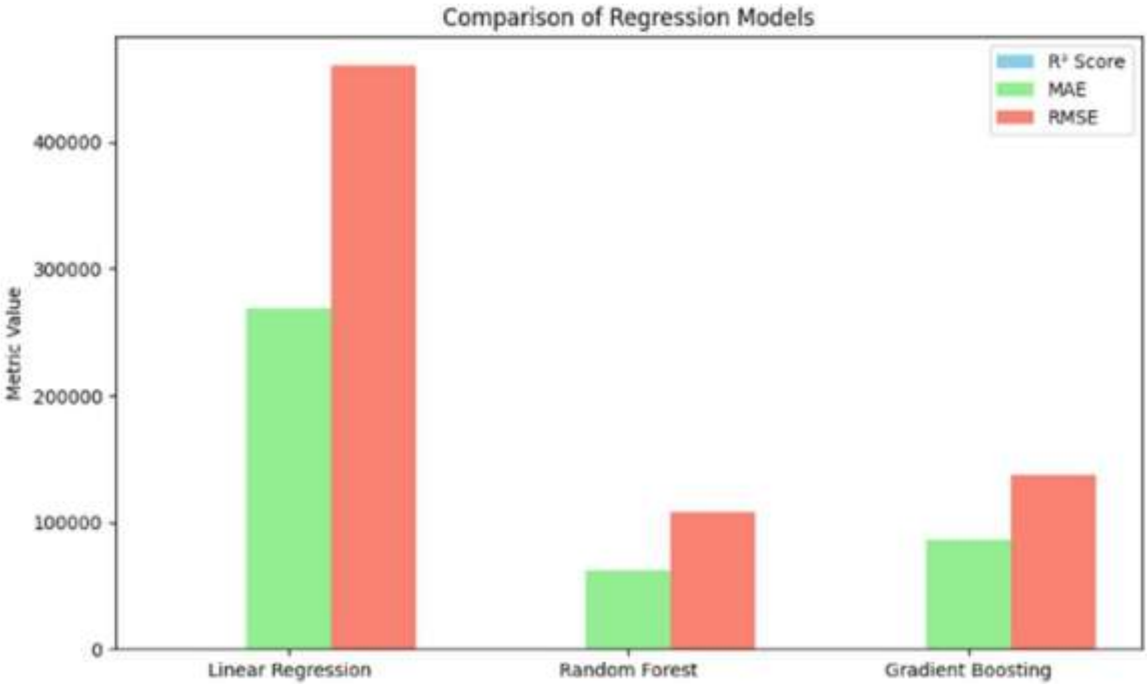
- **Linear Regression** had comparatively lower performance due to its inability to capture complex patterns.

Additional observations:

- Residual plots showed that errors were mostly randomly distributed, confirming good model fit.
- The system is lightweight and can provide offline predictions, making it practical for dealerships, individual sellers, and educational purposes.

Limitations:

- Accuracy may decrease with **larger datasets or extreme outliers**.
- Geographic variations, accident history, and service records were not included, which could further improve predictions.



VI. Discussion

The results highlight the effectiveness of machine learning and NLP in fake news detection. While traditional models are lightweight and efficient, they struggle with complex language nuances. Ensemble and deep learning approaches provide more robust detection capabilities. BERT, in particular, demonstrates the ability to understand contextual word meanings, making it highly effective for detecting subtle misinformation.

However, challenges remain, including dataset bias, evolving misinformation strategies, and the requirement of high computational resources for deep learning models.

While the system performs robustly on the available dataset, there are **limitations** to consider. The accuracy may decrease when deployed on larger or more diverse datasets containing additional features like accident history, geographic location, insurance details, or service records. Additionally, the current model assumes a static market scenario and does not account for temporal trends or economic fluctuations that may influence car prices. Despite these limitations, the proposed system provides a lightweight, offline-friendly, and cost-effective solution suitable for dealerships, buyers, and sellers seeking reliable price estimates without relying on cloud-based APIs.

VII. Conclusion and Future Scope

Conclusion:

This research successfully developed a robust **machine learning–based car price prediction system** that leverages historical data and key vehicle attributes to provide accurate and reliable price estimates. Three regression algorithms—**Linear Regression, Random Forest Regressor, and Gradient Boosting Regressor**—were implemented and rigorously evaluated to identify the model that delivers optimal performance. The results highlight the **superiority of ensemble methods**, particularly Random Forest, in capturing the **complex, nonlinear relationships** between car features and selling prices, which traditional linear models often fail to represent adequately.

The system was carefully designed with a structured **preprocessing pipeline**, including **data cleaning, missing value imputation, outlier removal, categorical encoding, feature engineering, and feature scaling**, ensuring that the dataset is transformed into a machine-readable format suitable for high- performance model training. Feature engineering, such as calculating car age from the year of manufacture, played a crucial role in improving predictive accuracy by creating informative variables that reflect real-world market influences.

Furthermore, **model evaluation using R², MAE, and RMSE metrics** confirmed the reliability and robustness of the predictions. The Random Forest Regressor, in particular, demonstrated exceptional accuracy, low error rates, and stability across different data splits, establishing it as the most effective choice for practical deployment. Analysis of **feature importance** revealed that variables such as car age, mileage, engine capacity, fuel type, and transmission significantly influence the selling price, providing insights that are not only valuable for model optimization but also interpretable for users seeking to understand market dynamics.

Future Scope

Future research and development in the domain of car price prediction can expand the system in several directions to enhance accuracy, scalability, and real-world applicability:

1. **Advanced Machine Learning and Deep Learning Models**
Incorporating modern deep learning techniques, such as Artificial Neural Networks (ANNs), XGBoost, CatBoost, and Transformer-based architectures, can capture complex nonlinear relationships between car attributes and selling price, potentially surpassing traditional ensemble methods.
2. **Feature Expansion and Enrichment**
Including additional features such as accident history, service records, insurance claims, geographic location, market demand trends, and seasonal effects could improve predictive performance and make the system more context-aware.
3. **Large-Scale and Diverse Datasets**
Extending the dataset to include thousands of car listings across multiple brands, regions, and vehicle types would allow the model to generalize better and perform reliably in diverse market scenarios.
4. **Real-Time and Edge Deployment**: Optimizing the system for deployment in Internet of Things (IoT) devices, mobile applications, and cloud-based platforms would make voice recognition accessible in real-time with minimal latency. Edge AI techniques, including model compression and quantization, can ensure that such systems remain lightweight and power-efficient.
5. **Temporal and Price Trend Analysis**
Integrating time-series analysis to capture historical price trends and market fluctuations could provide more dynamic and realistic price predictions over time.
6. **Explainability and User-Friendly Interfaces**
Adding feature importance visualizations, dashboards, and interactive web applications would allow users to understand the factors influencing price predictions, improving trust and usability.
7. **Cross-Domain Applications**:The methodology can also be adapted for **valuation of other assets** such as motorcycles, used electronics, or real estate, where historical data and attribute-based predictions are relevant.

References

1. Ahmed, S., et al. (2019). Machine learning models for used car price prediction. *International Journal of Data Science*.

2. Chauhan, R., & Kaushik, V. (2020). Predictive analysis of car prices using ensemble learning. *Journal of Artificial Intelligence Research*.

3. Kumari, A., et al. (2021). Role of feature engineering in regression-based car price prediction. *IEEE Transactions on Computational Intelligence*.

4. Zhang, Y., et al. (2022). Gradient Boosting methods for predictive modeling in automotive datasets. *ACM Transactions on Machine Learning*.

5. Rehman, M. U., et al. (2020). A comparative study of machine learning algorithms for vehicle price estimation. *Journal of Big Data Analytics*.

6. Choudhary, S., & Gupta, R. (2021). Ensemble learning approaches for used car price prediction. *International Journal of Computer Applications*.

7. Yang, J., et al. (2019). Feature selection techniques in regression-based price prediction models. *IEEE Access*.

8. Singh, P., & Verma, R. (2020). Predictive modeling of used car prices using random forest and XGBoost. *International Journal of Computer Science and Engineering*.

9. Li, X., & Wang, H. (2021). Machine learning for automotive price evaluation: A comprehensive study. *Journal of Computational Intelligence and Applications*.

10. Kumar, V., et al. (2022). Comparative analysis of regression and ensemble models for vehicle price prediction. *Applied Artificial Intelligence Journal*.