

Fake News Detection Report

Introduction:

Fake news has become a significant concern in today's information age, where the spread of false or misleading information can have severe consequences. To tackle this issue, machine learning techniques can be employed to build models that can identify and classify news articles as fake or genuine. This report presents a simple code implementation for fake news detection using Python and machine learning, specifically the Random Forest Classifier algorithm.

Methodology:

1. Data Preparation:

- The code begins by importing the necessary libraries, including pandas, scikit-learn's TfidfVectorizer, Random Forest Classifier, train_test_split, and accuracy score.
- A dataset is loaded using the `pd.read_csv()` function, assuming the file 'news_dataset.csv' contains the necessary data.
- The dataset is then split into features (X) and labels (y) to prepare for training and testing.

2. Training and Testing:

- The data is further split into training and testing sets using `train_test_split()`, with a test size of 20% and a random state of 42 for reproducibility.
- A TF-IDF vectorizer is created using `TfidfVectorizer()` to convert the textual data into numerical features. This vectorizer calculates the Term Frequency-Inverse

Document Frequency (TF-IDF) values for each word in the corpus.

- The training set is transformed using the vectorizer's ``fit_transform()`` function, while the testing set is transformed using ``transform()``.

- A Random Forest Classifier is initialized using ``RandomForestClassifier()``. This classifier is an ensemble algorithm that combines multiple decision trees to make predictions.

- The classifier is trained on the training set using ``fit()``, where it learns to map the input features to their corresponding labels.

3. Predictions and Evaluation:

- The trained classifier is used to make predictions on the test set by calling ``predict()`` on the transformed test features.

- The accuracy of the predictions is evaluated using the ``accuracy_score()`` function, which compares the predicted labels with the actual labels from the test set.

- Finally, the accuracy score is printed to the console using ``print()``.

Results:

The code provides a measure of accuracy, indicating how well the trained model performs in distinguishing fake and genuine news articles. The accuracy score is an important metric for evaluating the effectiveness of the model, with a higher score indicating better performance.

Conclusion:

In this report, a simple code implementation for fake news detection using Python and machine learning was presented. The code utilizes the Random Forest Classifier

algorithm to classify news articles as fake or genuine. However, it's important to note that this is a basic implementation, and real-world fake news detection systems require additional preprocessing steps, feature engineering, and model tuning for improved accuracy. Furthermore, the effectiveness of the model heavily relies on the quality and representativeness of the dataset used for training.

THANK YOU

TEAM NAME: Karunya

TEAM MEMBERS: Karunya N

Jothi V

Lavanya s