REPORT

ON

NEWS ARTICLE CLASSIFICATION

FAKE OR REAL

shutterstock.com · 2129561147

SUMITTED BY

YUVA SHNEE.G

Mini Project Report

1. Objective

The objective of this project is to build a Natural Language Processing (NLP) based machine learning model to classify news articles as Fake or Real. This is crucial in today's digital age where misinformation can spread rapidly. The model will take a news article as input and predict whether the content is genuine or fabricated.

2. Tools & Technologies

Python – Programming language

NLTK – Natural Language Toolkit for text preprocessing

Scikit-learn – ML algorithms and evaluation

Pandas – Data handling and manipulation

TF-IDF – Text vectorization

3. Dataset

Source: Kaggle Fake News Dataset

Description: Contains labeled news articles with features like title, text, subject, and label (1: Fake, 0: Real).

Size: ~20,000 records

4. Data Preprocessing (Using NLTK & Pandas)

1. Loading Data:df = pd.read_csv("news.csv")

2. Cleaning Text:Remove punctuation, stopwords, digits

Tokenize, lowercase, and lemmatize text

```
from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer

import re


stop_words = set(stopwords.words("english"))

lemmatizer = WordNetLemmatizer()
```

```
def clean_text(text):
  text = re.sub(r'[^a-zA-Z]', ' ', text)
    tokens = text.lower().split()
    tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in stop_words]
    return ' '.join(tokens)


df['text'] = df['text'].astype(str).apply(clean_text)
```

5. Feature Extraction (TF-IDF)

Convert text into numerical format using Term Frequency-Inverse Document Frequency.

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer(max_features=5000)
X = tfidf.fit_transform(df['text']).toarray()
y = df['label']
```

6. Model Training

Train a classification model using Logistic Regression and Naive Bayes.

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Logistic Regression
lr_model = LogisticRegression()
lr_model.fit(X_train, y_train)
# Naive Bayes
nb_model = MultinomialNB()
nb_model.fit(X_train, y_train)
```

7. Model Evaluation

Use Accuracy, F1 Score, and Confusion Matrix to evaluate model performance.

```
from sklearn.metrics import accuracy_score, f1_score

lr_pred = lr_model.predict(X_test)

nb_pred = nb_model.predict(X_test)

print("Logistic Regression Accuracy:", accuracy_score(y_test, lr_pred))

print("Naive Bayes Accuracy:", accuracy_score(y_test, nb_pred))

print("F1 Score (LR):", f1_score(y_test, lr_pred))
```

Typical Results:

Logistic Regression Accuracy: ~92%

Bayes Accuracy: ~89%

8.Conclusion

This project demonstrates how to apply Natural Language Processing and machine learning to solve a real-world problem – fake news detection. By leveraging logistic regression and Naive Bayes along with proper text preprocessing, the model achieves high accuracy.
