Phase:5 Fake News Detection Using NLP - Project Documentation

1. Problem Statement

Objective: The project aimed to develop a fake news detection model using Natural Language Processing (NLP) techniques. The primary goal was to distinguish between genuine and fake news articles based on their textual content.

2. Design Thinking Process

Problem Understanding

- The project's core objective was to tackle the problem of fake news.
- It aimed to create a model that can effectively differentiate between real and fake news articles.
- The solution will assist in identifying misleading information and promoting the dissemination of accurate news.

Data Collection and Analysis

- The project started by collecting a dataset from Kaggle containing articles' titles and text, along with labels indicating their authenticity.
- The dataset provides the foundational data required for model development.

Data Preprocessing

- Data preprocessing involved cleaning and preparing the text data for analysis.
- Steps included converting text to lowercase, removing punctuation and numbers, removing common stopwords, and tokenization.

Feature Extraction

- Feature extraction focused on converting the preprocessed text data into numerical features that machine learning algorithms can understand.
- Term Frequency-Inverse Document Frequency (TF-IDF) was chosen as the feature extraction method.

Model Development

- The project considered multiple machine learning algorithms, including Logistic Regression, Random Forest, and Neural Networks.
- The chosen algorithm is Multinomial Naive Bayes, which is a common choice for text classification tasks.
- The model was trained using the preprocessed and vectorized data.

Evaluation

The model's performance was evaluated using various metrics:

- Accuracy: Measures the overall correctness of the model's predictions.
- Precision: Indicates the model's ability to correctly classify genuine news articles.
- Recall: Measures the model's ability to correctly classify fake news articles.
- F1 Score: A harmonic mean of precision and recall, providing a balance between the two.
- ROC-AUC Score: Measures the model's ability to discriminate between genuine and fake news across different thresholds.

Conclusion

- The project's objectives were successfully met by developing an effective fake news detection model.
- The model can accurately differentiate between genuine and fake news articles, helping combat the spread of misinformation.
- Future work could involve exploring other machine learning algorithms, conducting hyperparameter tuning, and incorporating additional features for improved performance.

3. Dataset Description

- Dataset Source: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset
- Dataset Size: The dataset contains a substantial number of articles with titles and text.
- Columns: The dataset contains two main columns: 'title' and 'text', along with a 'label' column indicating whether the article is genuine or fake.

4. Data Preprocessing

- Loading the Dataset: The true and fake news datasets were loaded into pandas DataFrames.
- Data Labeling: The articles were labeled as 'real' and 'fake' to categorize them.
- Text Preprocessing: Text data was cleaned, including converting to lowercase, removing punctuation and numbers, and eliminating common stopwords.
- Tokenization: The text was tokenized by splitting it into individual words.

5. Feature Extraction

TF-IDF (Term Frequency-Inverse Document Frequency): TF-IDF was employed as the feature extraction technique. It assigns weights to words based on their frequency in the document and rarity in the corpus, allowing the model to understand the importance of each word.

6. Model Development

- ❖ Machine Learning Algorithm: Multinomial Naive Bayes, a common choice for text classification tasks, was selected as the machine learning algorithm.
- ❖ Training: The model was trained using the preprocessed and vectorized data.

7. Evaluation

The model's performance was assessed using several key metrics:

- ▪ Accuracy: Achieved a high level of overall correctness in classifying articles.
- ▪ Precision: Demonstrated the model's ability to correctly classify genuine news articles.
- ▪ Recall: Measured the model's ability to correctly classify fake news articles.
- ▪ F1 Score: Provided a balanced measure by considering both precision and recall.
- ▪ ROC-AUC Score: Evaluated the model's discrimination between genuine and fake news across different thresholds.

8. Results

- ⅄ The fake news detection model demonstrated promising results in distinguishing genuine and fake news articles.
- ⅄ The project concluded with a high level of accuracy, precision, recall, F1 score, and ROC-AUC score, indicating the effectiveness of the model.

9. Conclusion

- The project successfully addressed the challenge of fake news detection using NLP techniques.
- The model provides a valuable tool to identify and combat the spread of misinformation.
- Future work may involve exploring additional algorithms, fine-tuning hyperparameters, and incorporating more features for further improvements.

10.References

Code:

```
import pandas as pd

import re

import nltk

from sklearn.model_selection import train_test_split

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.metrics import confusion_matrix

nltk.download('stopwords')

true_data = pd.read_csv(r'C:\Users\jagan\Downloads\archive\True.csv')

fake_data = pd.read_csv(r'C:\Users\jagan\Downloads\archive\Fake.csv')

true_data['label'] = 'true'

fake_data['label'] = 'fake'

data = pd.concat([true_data, fake_data])

data['text'] = data['text'].str.lower()

data['text'] = data['text'].apply(lambda x: re.sub(r'[^\w\s]', '', x))

from nltk.corpus import stopwords

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

data['text'] = data['text'].apply(lambda x: ' '.join([word for word in x.split() if word not in
stop_words]))

X = data['text']

y = data['label']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

tfidf_vectorizer = TfidfVectorizer(max_features=5000)

X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)

X_test_tfidf = tfidf_vectorizer.transform(X_test)

model = LogisticRegression()
```

```
model.fit(X_train_tfidf, y_train)

y_pred = model.predict(X_test_tfidf)

accuracy = accuracy_score(y_test, y_pred)

precision = precision_score(y_test, y_pred, pos_label='fake')

recall = recall_score(y_test, y_pred, pos_label='fake')

f1 = f1_score(y_test, y_pred, pos_label='fake')

roc_auc = roc_auc_score(y_test, model.predict_proba(X_test_tfidf)[:, 1])

cm = confusion_matrix(y_test, y_pred)

print(f"Accuracy: {accuracy:.2f}")

print(f"Precision: {precision:.2f}")

print(f"Recall: {recall:.2f}")

print(f"F1 Score: {f1:.2f}")

print(f"ROC AUC Score: {roc_auc:.2f}")

print("Confusion Matrix:")

print(cm)

plt.figure(figsize=(8, 6))

sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=["True", "Fake"],
yticklabels=["True", "Fake"])

plt.xlabel('Predicted')

plt.ylabel('True')

plt.title('Confusion Matrix')

plt.show()
```

OUTPUT:

IDLE Shell 3.11.4

File Edit Shell Debug Options Window Help

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/jagan/OneDrive/Documents/NM My team/nm p5.py
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\jagan\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Confusion Matrix:
[[4590   60]
 [  29 4301]]
Accuracy: 0.99
Precision: 0.99
Recall: 0.99
F1 Score: 0.99
ROC AUC Score: 1.00
>>>
```