Fake News Prediction - Documentation

Project overview

This notebook performs a simple fake news classification pipeline on a text dataset. It demonstrates data loading, basic text preprocessing, feature extraction using TF-IDF, model training (Logistic Regression), and evaluation (accuracy). The notebook is intended for learning and quick prototyping.

Contents / Notebook flow

1. Imports & Setup - import required libraries and download NLTK resources.
2. Load dataset - read the CSV/TSV file containing news text and labels.
3. Exploratory checks - inspect shape, sample rows, class balance, missing values.
4. Preprocessing - clean text (remove punctuation, lowercasing), remove stopwords, stemming.
5. Feature extraction - convert text to numeric features using TfidfVectorizer.
6. Train / test split - split features and labels into training and testing sets.
7. Model training - train a LogisticRegression classifier.
8. Evaluation - compute accuracy and optionally other metrics (precision, recall, F1).
9. Save / export - save the trained model and vectorizer for later use.

How to run

1. Create a Python environment (recommended Python 3.8+).
2. Install required packages (see requirements.txt).
3. Open and run the notebook cells in order.

Requirements

pandas

numpy

scikit-learn

nltk

joblib

matplotlib

seaborn

Commented import block

```
# Numerical computing and array handling
import numpy as np  # fast numeric operations and arrays
# Data manipulation and I/O
import pandas as pd  # dataframe structures, reading/writing CSVs
# Regular expressions
import re  # text pattern cleaning
# NLTK
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
# Feature extraction
from sklearn.feature_extraction.text import TfidfVectorizer
# Model training and evaluation
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
# Saving model
import joblib
```

Recommended improvements

- Add lemmatization

- Include confusion matrix and F1 score

- Use pipelines

- Add hyperparameter tuning

End of documentation.