

Fake News Detection Project Report

Introduction

I built a fake news detection system for my internship using machine learning, and I'm really proud of how it turned out! The goal was to take a dataset of news headlines and figure out if they're real (0) or fake (1). The dataset had two columns: "text" (the headlines) and "label" (0 or 1). I did this project in Google Colab with Python, and it was my first time working on something like this.

My Approach

Here's how I tackled the project, step by step:

1. **Loading the Data:** I used pandas to load the dataset. It had many unnecessary columns and I dropped all except text, then I added another column "label", for true it indicates 1 and for fake it indicates 0, a "text" column with headlines and a "label" column showing if they're real or fake. I concatenated both csv files and reshuffled the data. I checked for missing data and found none, so I was ready to move forward.
2. **Cleaning the Text:** I wrote a function called `clean_text` to clean up the headlines:
 - Made everything lowercase (e.g., "LONDON" → "london").
 - Removed punctuation like commas and parentheses.
 - Split the text into words using NLTK's `word_tokenize`.
 - Took out common words like "the" and "is" (called stopwords).
 - Simplified words to their root form (e.g., "running" → "run") using lemmatization. I saved the cleaned text in a new column called `cleaned_text`.
3. **Turning Text into Numbers:** I used `TfidfVectorizer` from scikit-learn to turn the cleaned text into numbers that a model can understand. I limited it to 5000 features to keep things manageable.
4. **Training the Model:** I split the data into 80% for training and 20% for testing. Then, I trained a Naïve Bayes classifier (`MultinomialNB`) because it's good for text tasks like this.
5. **Testing the Model:** I checked how well the model worked on the test set and even tried it on a made-up headline: "Breaking: Aliens land in London!"
6. **Saving the Model:** I saved the model and vectorizer using `joblib` so I can use them again later.

Challenges I Faced

- **NLTK Error:** I ran into a `LookupError: Resource punkt_tab not found` error when cleaning the text. I fixed it by downloading the missing NLTK resource with `nltk.download('punkt_tab')`.
- **Understanding the Data:** I had to double-check what the labels meant (1 for real, 0 for fake) by looking at the dataset's documentation.
- **Text Formats:** The headlines had different styles (like "LONDON (Reuters)"), so I had to make sure my cleaning function handled them consistently.

How Well It Worked

My model did awesome! Here's what I got:

- **Accuracy:** 92.96% (so it got 92.96% of the predictions right).
- **Detailed Results:**

text

CollapseWrapCopy

	precision	recall	f1-score	support
0	0.93	0.94	0.93	4698
1	0.93	0.92	0.93	4282
accuracy			0.93	8980

This means the model was really good at spotting both real and fake news, with balanced precision and recall.

- **Sample Prediction:**
 - Input: "Breaking: Aliens land in London!"
 - Output: "Fake" (which makes sense since it sounds like a wild story!).

Improvements

Honestly, with an accuracy of 92.96%, I didn't feel the need to improve the model further—it's already working great! If I had more time, I might try cleaning the text even more (like removing numbers or URLs) or testing other models like Random Forest, but I'm happy with these results for now.

Conclusion

I successfully built a fake news detection system using a Naïve Bayes classifier, and it works really well with 92.96% accuracy. I learned a ton about cleaning text, turning it into numbers, and training a model. Even though I hit a few bumps (like the NLTK error), I figured them out and got a solid result. This project was a great way to dip my toes into machine learning, and I'm excited to do more!