FactLens: Misinformation Detection

CMPSC 463 (section 1) **GitHub** **Avik Bhuiyan & Mostafa Amer**

Project Description

The goal of FactLens is to be able to detect false information in online articles, journals, and research papers. It does this by using a supervised classifier along with an unsupervised theme extractor. FactLens is a web application that allows users to input any text from an article, then analyzes the information and displays results. This includes information such as the probabilities of being true or false, cluster themes, cluster sizes, and top terms.

Significance

Information on the internet is not guaranteed to be factual and is a major problem today. When citing or using information found online, it can be difficult to determine if it is FactLens solves this by:

- Automatically detects text that is likely to be false
- Fetches cluster themes found in the data
- Displays meaningful information on the analyzed data and probability scores

FactLens is an easy-to-use, single-page, application that can be utilized by many people such as students, researchers, teachers, and just the general population who browse the internet to fact check information.

Code Structure

The project files are structured in a typical flask web application. The Flask backend is served by running the app.py and the frontend is viewable via the address http://127.0.0.1:8000.

```
FactLens/
    Data/                   /* contains dataset files */
    Models/                 /* saved ML models */
    Webapp/                 /* folder containing frontend and backend */
        static/             /* styling and logic for API requests and UI changes */
        template/           /* frontend HTML */
    app.py              /* main Flask web API */
```

Train_models.py /* pipeline for training classifier and clustering */

Functionalities and Test Results

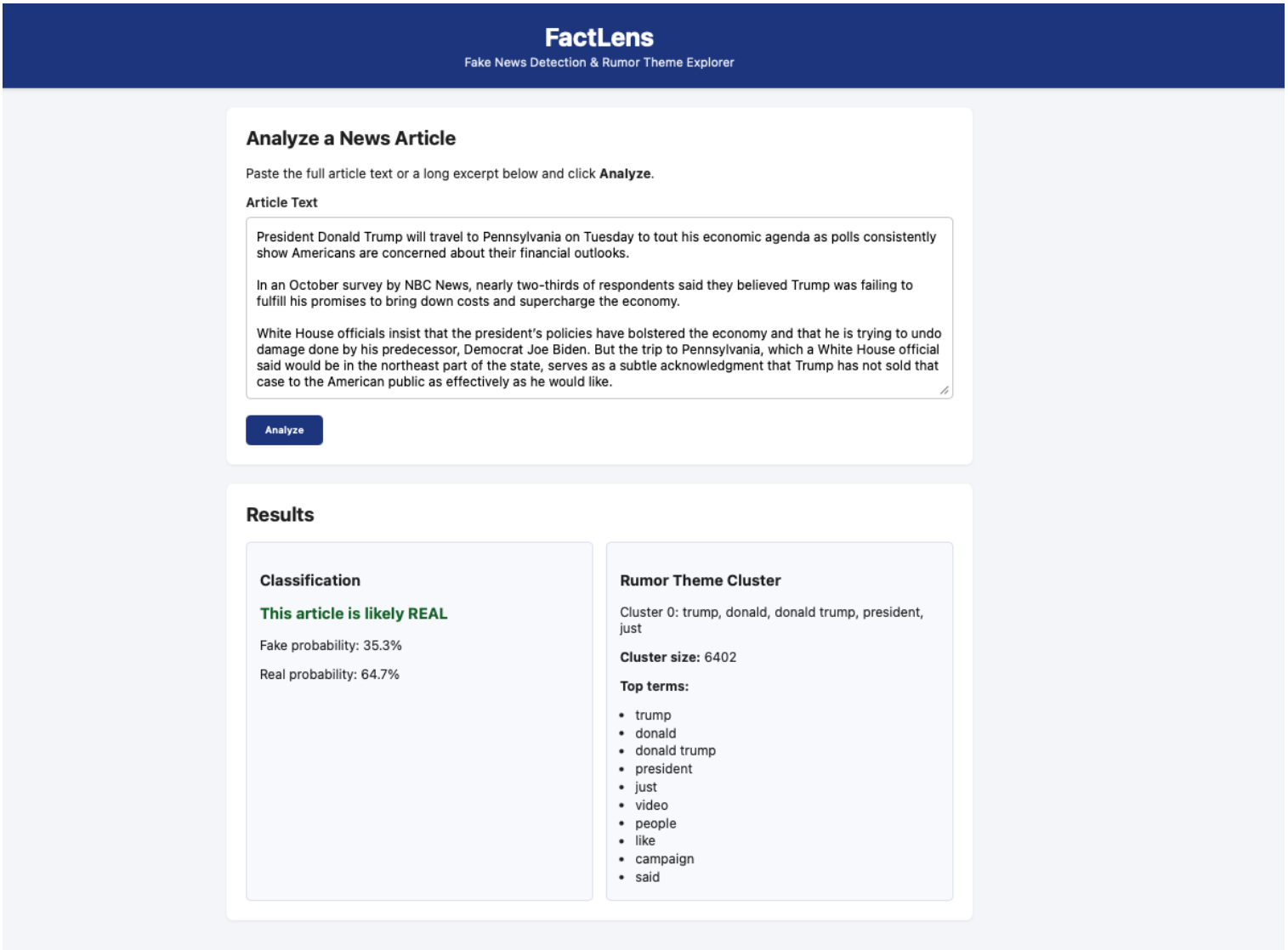
FactLens contains these main features:

1. **A Fake News Classifier**

   This classifier uses TF-IDF and a Logistic Regression model to determine if an article is real or fake. The TF-IDF model converts text into numerical features, allowing the system to capture important words. The logistic regression on the other hand learned using the data and returns probabilities on whether it is fake or real, and an overall indicator of the data.

2. **Rumor Theme Clustering**

   The clustering feature is able to distinguish themes in the given article. The data is converted to TF-IDF vectors, then PCA is applied to reduce its dimensionality, and lastly K-Means is used to group the text into different clusters.



Data Collection

The data used for our models was obtained from the **Kaggle Fake and Real News Dataset** via https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset. Its contents, including the .csv files, can be found in the /data/ folder. They contain a vast number of articles with data such as title and text and is used in our modeling.

Data Preprocessing and Feature Engineering

Our **preprocessing** pipeline consists of multiple processes. This includes:

- Filling in missing fields
- Merging some columns into features
- Removing articles out of our desired range
- TF-IDF vectorization

These steps are performed after collecting the data by running the train_models.py file. Additionally, we used **feature engineering** to boost the performance of our models.

For classification:

- N-grams to find word sequences for sentence context

For clustering:

- TF-IDF to transform text into vectors
- PCA to reduce dimensionality and prepare for K-Means

Model Development

FactLens involves two different machine learning models, one **supervised** and the other **unsupervised**, and both built using the Kaggle News Dataset.

For the supervised classification, TF-IDF vectorization is used and converts text into numbers. This keeps track of significant words while also removing stop words. We use a logistic regression model since they are ideal for text classification. We also perform a GridSearchCV, testing different TF-IDF sizes to optimize the model, accurately determining if an article is real or fake.

Additionally, FactLens also features an unsupervised clustering model, using TF-IDF to reduce the data, PCA for reducing the dimensionality of the news data. then performing K-Means to

cluster the information. The system takes keywords from each cluster and uses it to characterize that group's theme.

Conclusion

Overall, FactLens demonstrates skills we learned throughout the course, as well as solutions for the increasing issue of online misinformation. FactLens analyzes article text and returns whether it is likely fake or real along with cluster themes within the data. We combined a supervised and unsupervised model to give us further insight into if a text is real or fake.

The classifier uses TF-IDF features and a logistic regression and the clustering uses PCA and K-Means. We implemented data preprocessing, feature engineering, model tuning, and evaluated the results, in which FactLens was successful in its goal of detecting true and false data. If we had more time and could improve the application, we would have used a larger dataset to get more accurate results and improved our models by incorporating more advanced methods. The frontend also could show more information including bar charts or signify the cluster keywords more.