Project Report: Classification of Faculty directory pages and homepages

Overview:

ExpertSearch System is a search engine that is developed by students of University of Illinois to facilitate faculty search based on their research interests. The goal for this project is to enhance or aid the ExpertSearch System by writing a classification utility to identify good faculty directory and faculty webpage URLs. This utility/functionality can be added to the existing search system later on.

Goals:

Goals for this project are

1. Identifying faculty directory URLs
2. Identifying faculty webpage URLs

Proposed Solution:

Following methodology will be used as a starting point to achieve above stated goals.

1. Use data provided in Assignment MP2.1 and MP2.3 labeled as good examples and collect data for bad URLs and classify these as bad examples.
2. Based on this data, scrape these URLs to pre-process data.
3. Automatic or manual feature extraction based on data collected in step 2.
4. Develop a utility in Python to identify good URLs from bad ones.

Data Collection:

Data set used for this project was provided partially under Assignments MP2.1 and MP2.3. I used following steps for data collection.

1. Collect URLs that can be labeled as NonFaculty.
2. Append FacultyDirectory URLs to this.
3. Append Faculty URLs to this.
4. Scrape these URLs
5. Apply data cleaning and pre-processing.
6. Remove tags to get text.
7. Remove Whitespaces & unwanted characters.
8. Remove Stop Words, Apply Stemming and Lemmatizing using NLTK
9. Replace email addresses and Web addresses with text tags
10. Keep the desired data and store in csv for later use.

This step is done in DataLoad.ipynb available in github project directory.
Input : *LoadSheet.csv (available in github project directory.)*
Output : *extracted_data_unprocessed_latest.csv , extracted_data_processed.csv (Both available in github project directory)*

Feature Extraction & Model Building:

Using the labeled data from previous step, I tried multiple methods for feature extraction. Following features were extracted initially.

- No. of times professor, assistant, lecturer appears in text.
- No of times faculty, staff and people appear in text.
- No of times university, institute, school, department appear in text.
- No of times email addresses appear in text.
- No of times Journal, Research, Publications appear in text.

I was not able to fit model successfully with these features. I believe that this needs to be further worked on and will assist in multi-class classification.

I used sklearn library to extract features based on tokenization and TF-IDF Vectorization and fit a prediction model. I tried multiple classification models like Simple Bag-of-words model, Naive Bayes, logistic regression etc. I also explored multiple libraries to achieve this task like GenSim, PyTorch.

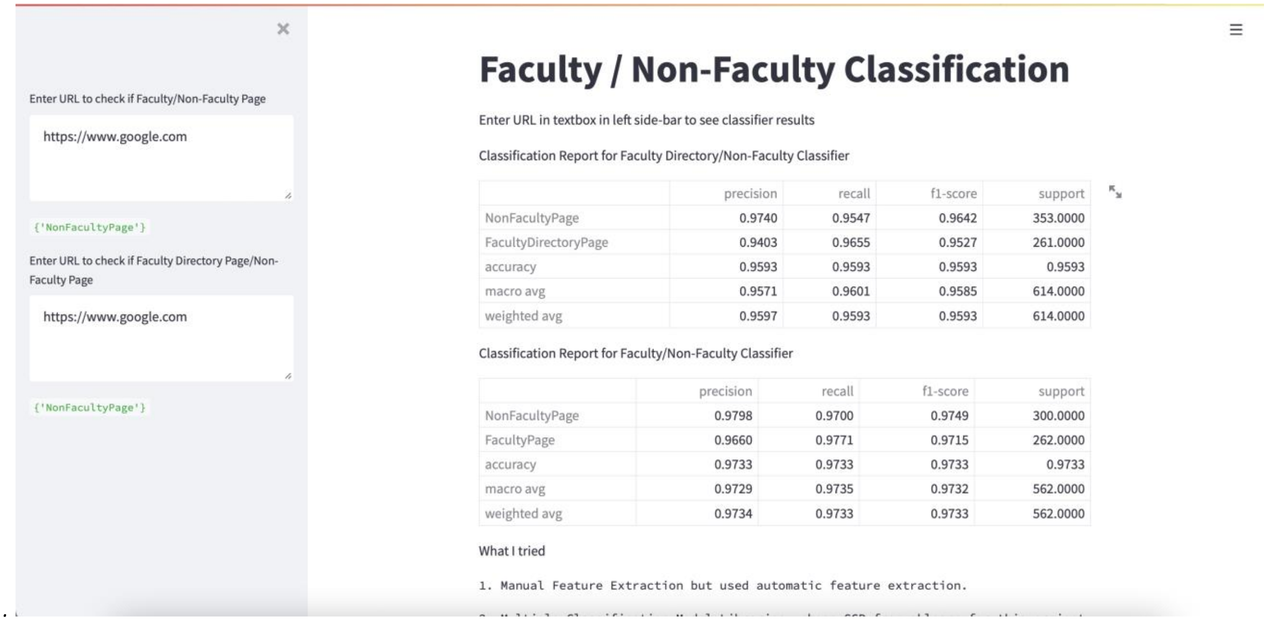
Two models were fit.

1. Classify faculty directory URLs from Non Faculty URLs
2. Classify faculty webpage URLs from Non Faculty URLs

This step is done in Feature Extraction&ModelBuilding.ipynb.ipynb available in github project directory.
Input : *bios.txt, extracted_data_processed.csv (available in github project directory.)* ***Output :*** *clf_rptFac.csv, clf_rptFacDir.csv, Fac_classifier & FacDir_classifier models (Both available in github project directory)*

A Simple GUI:

A simple GUI was built based on the outputs from previous step. User can input URL and it will show result of classification algorithms.



Enter URL to check if Faculty/Non-Faculty Page

https://www.google.com

{'NonFacultyPage'}

Enter URL to check if Faculty Directory Page/Non-Faculty Page

https://www.google.com

{'NonFacultyPage'}

Faculty / Non-Faculty Classification

Enter URL in textbox in left side-bar to see classifier results

Classification Report for Faculty Directory/Non-Faculty Classifier

	precision	recall	f1-score	support
NonFacultyPage	0.9740	0.9547	0.9642	353.0000
FacultyDirectoryPage	0.9403	0.9655	0.9527	261.0000
accuracy	0.9593	0.9593	0.9593	0.9593
macro avg	0.9571	0.9601	0.9585	614.0000
weighted avg	0.9597	0.9593	0.9593	614.0000

Classification Report for Faculty/Non-Faculty Classifier

	precision	recall	f1-score	support
NonFacultyPage	0.9798	0.9700	0.9749	300.0000
FacultyPage	0.9660	0.9771	0.9715	262.0000
accuracy	0.9733	0.9733	0.9733	0.9733
macro avg	0.9729	0.9735	0.9732	562.0000
weighted avg	0.9734	0.9733	0.9733	562.0000

What I tried

```
1. Manual Feature Extraction but used automatic feature extraction.
```

Libraries & Environment Specification:

This project was built on Anaconda and Python3.7. This was built using MACBOOK AIR M1 2020. Following libraries will need to be installed in order to execute this.

1. NLTK
2. SKLearn
3. Streamlit
4. BeautifulSoup

Installation commands for these libraries and instructional video tutorial are available in project directory

Accuracy and Performance:

For both classifiers, accuracy greater than 91% was achieved.

Challenges & Interesting Findings

- Data Gathering is extremely time-consuming. Around 60% of my time was spent on this task alone.
- Gensim TFIDF Vectorizer output was quite strange for this dataset. Unfortunely, did not have enough time to further explore this.
- Skewed Dataset almost always favors in prediction of majority class.

Conclusion

This was a very interesting project for me. Since I am very new to Python and Data Mining field, I learned a lot about different models and was able to implement a couple of techniques as well learned during this course.