STEVENS Institute of Technology

BBC News Document Classification

Group 8

Varnika Toshniwal(10473454), Yogesh Awdhut Gadade (10467214), Patrick Cullinane (10473527)

Content

1. Intro, tools and frameworks (by Yogesh)
2. Data Preprocessing (by Yogesh)
3. Data Insights (by Varnika)
4. Models: LinearSVC, Naive Bayes (by Yogesh)
5. Models: RandomForest (by Varnika)
6. Models: XGBoost (by Patrick)
7. Model Selection (by Patrick)
8. Conclusion (by Patrick)

1. Introduction

Introduction

- Problem: BBC news document Multiclass Classification problem
- Dataset: BBC News dataset (Unstructured text data)
- Explored text data processing techniques
- Explored ML solutions for the Multiclass Classification problem
  - Implemented MNB, Linear SVM, RF and XGBoost

Tools and Frameworks

- Github repository
  - https://github.com/cullinap/cpe695-project
- Anaconda-Jupyter notebook
  - Python 3.6
  - Python libraries such as nltk, sklearn, matplotlib, pandas, numpy etc
- Communication and cooperation:
  - Google Docs, Slides, Drive, Zoom for cooperation
  - Latex on Overleaf

2. Data, Data Preprocessing

BBC News Dataset

- Each class folder contains txt files

Ad sales boost Time Warner profit

Quarterly profits at US media giant TimeWarner jumped 76% to $1.13bn (£600m) for the three months to December, from $639m year-earlier.

The firm, which is now one of the biggest investors in Google, benefited from sales of high-speed internet connections and higher advert sales. TimeWarner said fourth quarter sales rose 2% to $11.1bn from $10.9bn. Its profits were buoyed by one-off gains which offset a profit dip at Warner Bros, and less users for AOL.

Time Warner said on Friday that it now owns 8% of search-engine Google. But its own internet business, AOL, had has mixed fortunes. It lost 464,000 subscribers in the fourth quarter profits were lower than in the preceding three quarters. However, the company said AOL's underlying profit before exceptional items rose 8% on the back of stronger internet advertising revenues. It hopes to increase subscribers by offering the online service free to TimeWarner internet customers and will try to sign up AOL's existing customers for high-speed broadband. TimeWarner also has to restate 2000 and 2003 results following a probe by the US Securities Exchange Commission (SEC), which is close to concluding.

Time Warner's fourth quarter profits were slightly better than analysts' expectations. But its film division saw profits slump 27% to $284m, helped by box-office flops Alexander and Catwoman, a sharp contrast to year-earlier, when the third and final film in the Lord of the Rings trilogy boosted results. For the full-year, TimeWarner posted a profit of $3.36bn, up 27% from its 2003 performance, while revenues grew 6.4% to $42.09bn. "Our financial performance was strong, meeting or exceeding all of our full-year objectives and greatly enhancing our flexibility," chairman and chief executive Richard Parsons said. For 2005, TimeWarner is projecting operating earnings growth of around 5%, and also expects higher revenue and wider profit margins.

TimeWarner is to restate its accounts as part of efforts to resolve an inquiry into AOL by US market regulators. It has already offered to pay $300m to settle charges, in a deal that is under review by the SEC. The company said it was unable to estimate the amount it needed to set aside for legal reserves, which it previously set at $500m. It intends to adjust the way it accounts for a deal with German music publisher Bertelsmann's purchase of a stake in AOL Europe, which it had reported as advertising revenue. It will now book the sale of its stake in AOL Europe as a loss on the value of that stake.

BBC News Dataset - Collected and Labeled

	NewsText	NewsType
0	Ad sales boost Time Warner profit\n\nQuarterly...	business
1	Dollar gains on Greenspan speech\n\nThe dollar...	business
2	Yukos unit buyer faces loan claim\n\nThe owner...	business
3	High fuel prices hit BA's profits\n\nBritish A...	business
4	Pernod takeover talk lifts Domecq\n\nShares in...	business

Pre-processing implemented

- Basic filters applied - removed null and duplicate values if any
- Processed input News Text samples using text processing methods
- Since Machine Learning model understands numbers hence
  - vectorised the processed news text
- Label encoding - one-hot encoding

```
{0: 'business', 1: 'entertainment', 2: 'politics', 3: 'sport', 4: 'tech'}
```

Preparing text data

```
1 sampleText
```

```
'The messages will be "unwrapped" by sculptor Richard Wentworth, who is responsible for decorating the tree with broken plates and light bulbs. Artists who have decorated the Tate tree in previous years include Tracey Emin in 2002.'
```

```
1 preprocessDataset(sampleText)
```

```
--- Tokenized ----
 ['the', 'messages', 'will', 'be', 'unwrapped', 'by', 'sculptor', 'richard', 'wentworth', 'who', 'is', 'responsible', 'for', 'decorating', 'the', 'tree', 'with', 'broken', 'plates', 'and', 'light', 'bulbs', 'artists', 'who', 'have', 'decorated', 'the', 'tate', 'tree', 'in', 'previous', 'years', 'include', 'tracey', 'emin', 'in', '2002']

---- Removed Stop Words ----
  ['messages', 'unwrapped', 'sculptor', 'richard', 'wentworth', 'responsible', 'decorating', 'tree', 'broken', 'plates', 'light', 'bulbs', 'artists', 'decorated', 'tate', 'tree', 'previous', 'years', 'include', 'tracey', 'emin', '2002']

---- Joined as Sentense ----
 messages unwrapped sculptor richard wentworth responsible decorating tree broken plates light bulbs artists decorated tate tree previous years include tracey emin 2002

---- Numbers removed ----
 messages unwrapped sculptor richard wentworth responsible decorating tree broken plates light bulbs artists decorated tate tree previous years include tracey emin

---- Stemmed ----
 messag unwrap sculptor richard wentworth respons decor tree broken plate light bulb artist decor tate tree previou year includ tracey emin

---- Lemmatized ----
 messag unwrap sculptor richard wentworth respons decor tree broken plate light bulb artist decor tate tree previou year includ tracey emin
```

Preparing Text data - vectorization

- Converted processed text data into machine understandable numbers using TF-IDF: **Term Frequency-Inverse document frequency**
- TF = (# of reparations of word in a document) / (# of words in a document)
- IDF=Log(# of doc.s) / (# of docs containing the word)
- Vectorized to create matrix of all text samples

Multinomial Naive Bayes

- Multinomial naïve Bayes: With a multinomial event model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial (p1, p2, …, pn) where pi is the probability that event i occurs. A feature vector x=(x1, x2, …, xn) is then a histogram, with xi counting the number of times event i was observed in a particular instance.

- Why this algorithm for the BBC News classification?

BBC News Classification MNB Results

accuracy: 0.9553990610328639

Classification report:

	precision	recall	f1-score	support
0	0.94	0.99	0.96	97
1	0.98	0.89	0.94	73
2	0.94	0.97	0.96	78
3	0.95	1.00	0.97	105
4	0.98	0.89	0.94	73
accuracy			0.96	426
macro avg	0.96	0.95	0.95	426
weighted avg	0.96	0.96	0.95	426

Confusion matrix:

```
[[ 96   0   0   0   1]
 [  3  65   4   1   0]
 [  1   0  76   1   0]
 [  0   0   0 105   0]
 [  2   1   1   4  65]]
```

```
{0: 'business', 1: 'entertainment', 2: 'politics', 3: 'sport', 4: 'tech'}
```

Linear Support Vector Machine Classifier using libsvm

- What is Linear SVC?

- Why this algorithm for the BBC News classification?

BBC News Classification Linear SVC Results

accuracy: 0.9812206572769953

Classification report:

	precision	recall	f1-score	support
0	1.00	0.98	0.99	97
1	0.97	0.99	0.98	73
2	0.99	0.97	0.98	78
3	0.97	1.00	0.99	105
4	0.97	0.96	0.97	73
accuracy			0.98	426
macro avg	0.98	0.98	0.98	426
weighted avg	0.98	0.98	0.98	426

Confusion matrix:

```
[[ 95   0   0   1   1]
 [  0  72   1   0   0]
 [  0   0  76   1   1]
 [  0   0   0 105   0]
 [  0   2   0   1  70]]
```

{0: 'business', 1: 'entertainment', 2: 'politics', 3: 'sport', 4: 'tech'}

3. Data insights

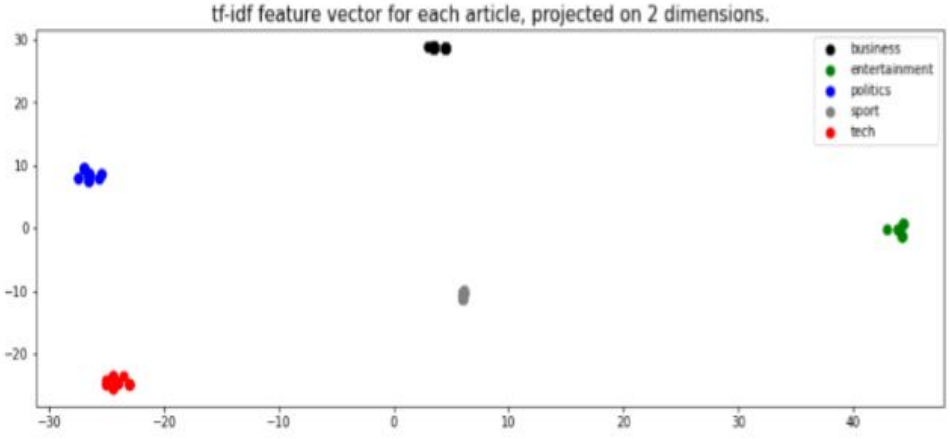

Fig. 5. For each category, find words that are highly corelated to it

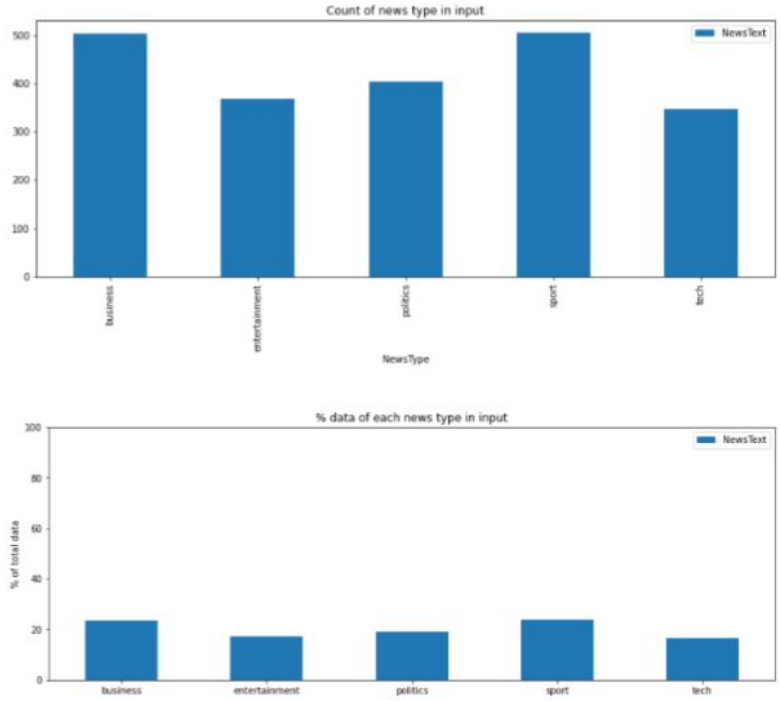

Fig. 3. Data representation

```
# Category: 'business':
  . Most correlated unigrams:
       . politics
       . sport
       . business
  . Most correlated bigrams:
       .
# Category: 'entertainment':
  . Most correlated unigrams:
       . business
       . sport
       . entertainment
  . Most correlated bigrams:
       .
```

```
# Category: 'politics':
  . Most correlated unigrams:
       . business
       . sport
       . politics
  . Most correlated bigrams:
       .
# Category: 'sport':
  . Most correlated unigrams:
       . politics
       . business
       . sport
  . Most correlated bigrams:
       .
# Category: 'tech':
  . Most correlated unigrams:
       . business
       . sport
       . tech
  . Most correlated bigrams:
       .
```

Fig. 4. For each category, find words that are highly corelated to it

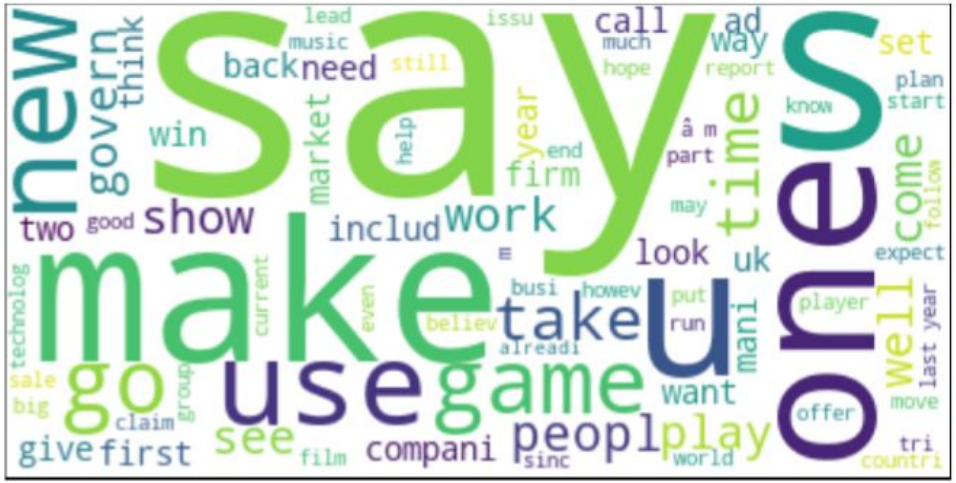

Fig. 6. word cloud of top 80 words from the corpus

3. Model Details

Random Forest Classifier

- A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
- For classification tasks, the output of the random forest is the class selected by most trees.
- An important motivation for using RF was the application of late fusion strategies based on the RF operational capabilities

Random Forest Model Results

accuracy: 0.9272300469483568

Classification report:

	precision	recall	f1-score	support
0	0.90	0.95	0.93	107
1	0.92	0.88	0.90	78
2	0.97	0.94	0.96	71
3	0.90	0.98	0.94	103
4	0.98	0.84	0.90	67
accuracy			0.93	426
macro avg	0.94	0.92	0.93	426
weighted avg	0.93	0.93	0.93	426

Confusion matrix:

```
[[102   1   1   2   1]
[  6  69   0   3   0]
[  2   0  67   2   0]
[  1   0   1 101   0]
[  2   5   0   4  56]]
```

XGBoost

	precision	recall	f1-score	support
0	0.95	0.96	0.96	108
1	0.93	0.93	0.93	61
2	0.97	0.92	0.94	91
3	0.97	0.99	0.98	97
4	0.94	0.96	0.95	69
accuracy			0.96	426
macro avg	0.95	0.95	0.95	426
weighted avg	0.96	0.96	0.96	426

Confusion matrix:

```
[[104   2   1   0   1]
 [  1  57   2   0   1]
 [  1   2  84   2   2]
 [  1   0   0  96   0]
 [  2   0   0   1  66]]
```

- Gradient Boosting: Add predictors to an ensemble with each subsequent model making corrections to the last. Make a prediction, calculate the residual, fit the next model to the residual for a specified amount of times or until residual is too small (1)
- XGB adds in a regularized term that prunes the trees and allows the model to generalize better (2)
- Model was run out of the box and with hyperparameter optimization just optimizing max_depth (due to computation constraints).
- “The final prediction for a given example is the sum of predictions from each tree.”
  - See: XGBoost: A Scalable Tree Boosting System by Chen et. al.

1. Hands On Machine Learning: chapter 7
2. Statquest video: xgboost https://www.youtube.com/watch?v=8b1JEDvenQU

Model Development

```
# train/test/split
X_train, X_test, y_train, y_test = train_test_split(
                                                X,y, test_size=0.2
                                            )
```

- SVM and NB
  - Parameter tuning
- Grid search for XGBoost
  - Only tuned max_depth in this project

- Labels encoding
- train/test split: 80/20: *support (426)*
  - Validation set (if used) 3 fold for param tuning on xgboost (again small due to comp constraints)
- Consider text preprocessing
  - TFIDF: min_df (ignore terms appearing < 5 docs)

```
{0: 'business', 1: 'entertainment', 2: 'politics', 3: 'sport', 4: 'tech'}
```

Metrics

```
avior.
accuracy: 0.9553990610328639

Classification report:

              precision    recall  f1-score   support

           0       0.95      0.96      0.96       108
           1       0.93      0.93      0.93        61
           2       0.97      0.92      0.94        91
           3       0.97      0.99      0.98        97
           4       0.94      0.96      0.95        69

    accuracy                           0.96       426
   macro avg       0.95      0.95      0.95       426
weighted avg       0.96      0.96      0.96       426


Confusion matrix:

 [[104   2   1   0   1]
 [  1  57   2   0   1]
 [  1   2  84   2   2]
 [  1   0   0  96   0]
 [  2   0   0   1  66]]
```

- Collected precision, recall, accuracy, f1 scores for each model
- Output confusion matrix and misclassified text for each model

```
Row 172 has been classified as sport and should be  tech
Row 188 has been classified as business and should be  tech
Row 253 has been classified as business and should be  tech
```

Overall Results

```
accuracy: 0.9812206572769953

Classification report:

              precision    recall  f1-score   support

           0       1.00      0.98      0.99        97
           1       0.97      0.99      0.98        73
           2       0.99      0.97      0.98        78
           3       0.97      1.00      0.99       105
           4       0.97      0.96      0.97        73

    accuracy                           0.98       426
   macro avg       0.98      0.98      0.98       426
weighted avg       0.98      0.98      0.98       426


Confusion matrix:

 [[ 95   0   0   1   1]
 [  0  72   1   0   0]
 [  0   0  76   1   1]
 [  0   0   0 105   0]
 [  0   2   0   1  70]]
```

- It appears that linearSVC performed best as compared to Naive Bayes, Random Forest, and XGBoost on these data.
- LinearSVC misclassified very few examples: (see confusion matrix to the left)
- No preference for precision OR recall in this case we chose to maximize f1 score

Conclusion

- Found that in the narrow case of text classification it appeared that linearSVC performed the best. We did not consider performance or the implications or adding other features in a wider context.
- Further research can be conducted on text vectorization by use of other methods such as word2vec. Additionally, tweaking ngrams, stopwords, and other language parameters.
- Can extend this problem to use-cases like a chatbot or classifying deduplicated text. Also train on a bigger dataset.
- Explore methods to further tune the models using various hyperparameter optimization techniques.
- Use a neural network model such as BERT and compare the results against our “traditional” ML models.