Klasyfikacja fałszywych wiadomości za pomocą NLP

Adam Duchna

1. Wstęp

   Projekt zajmie się problemem klasyfikacji wiadomości na podstawie częstotliwości występowania przetworzonych wcześniej słów zawartych zarówno w nagłówkach, jak i treści, a następnie oceną działania użytych klasyfikatorów na nagłówkach i treści.

2. Przetworzenie danych

   Przed wcześniejszym wykorzystaniem dane należało odpowiednio przetworzyć. Po wczytaniu dodano binarną etykietę informującą o fałszywości (0), bądź prawdziwości (1) informacji. Następnie dane złączono i przemieszano w celu uzyskania sensownej próbki do danych treningowych i testowych.

```
fake_news = pd.read_csv('Fake.csv')
true_news = pd.read_csv('True.csv')

fake_news['label'] = 0
true_news['label'] = 1

news = pd.concat([fake_news, true_news])
news = news.reset_index(drop=True)

news_title = news.drop(['subject', 'date', 'text'], axis=1)
news_text = news.drop(['subject', 'date','title'], axis=1)

news_title = shuffle(news_title, random_state=275035)
news_text = shuffle(news_text, random_state=275035)

news_title['title'] = news_title['title'].apply(cleanData)
news_title.to_csv('title_data.csv')

news_text['text'] = news_text['text'].apply(cleanData)
news_text.to_csv('text_data.csv')
```

Ostatnim krokiem było jeszcze przetworzenie samej treści tj. nagłówka i zawartości wiadomości. W tym celu powstała poniższa funkcja:

```
def cleanData(text):
    wordnet_lemmatizer = WordNetLemmatizer()
    stop_words = stopwords.words('english')
    stop_words.append('')
    text = text.lower()
    text = ''.join(c for c in text if not c.isdigit())
    text = ''.join(c for c in text if c not in punctuation)
    text = ' '.join([w for w in nltk.word_tokenize(text) if w not in stop_words])
    text = [wordnet_lemmatizer.lemmatize(word) for word in nltk.word_tokenize(text)]
    text = " ".join(w for w in text)
    return text
```

Eliminuje ona wszystkie dane liczbowe, znaki interpunkcyjne oraz słowa zawarte w stop liście języka angielskiego, a następnie lemetyzuje pozostałe dane w celu uogólnienia ich postaci.

3. Klasyfikacja i jej wyniki

Wszystkie poniżej powstałe klasyfikatory zostały stworzone za pomocą macierzy tfidf (odwrotnej częstości wyrażeń w dokumentach), dane treningowe i testowe stanowią odpowiednio 70% i 30% całego zbioru danych.

Wyniki klasyfikacji dla nagłówków

Classifier	Accuracy	Properly Classified	Misclassified
Linear regression	94.20%	12,689	781
Linear SVC	93.99%	12,660	810
Passive-Aggressive	93.88%	12,645	825
Naive Bayes	93.61%	12,609	861
Random Forest	92.04%	12,398	1,072
Decision tree	90.43%	12,181	1,289
K-Neighbors	89.08%	11,999	1,471

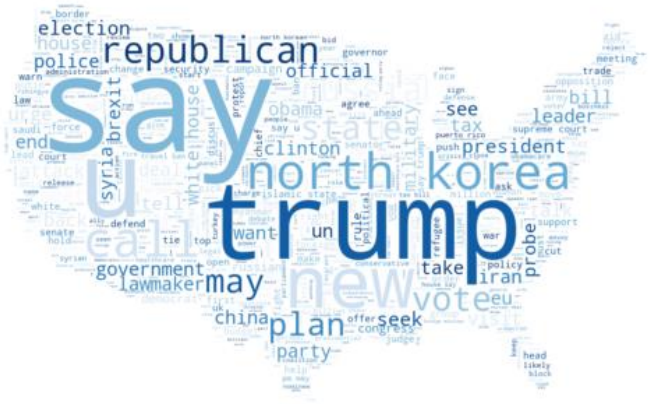
Wyniki klasyfikacji dla treści

Classifier	Accuracy	Properly Classified	Misclassified
Decision tree	99.65%	13,233	47
Linear SVC	99.27%	13,183	97
Passive-Aggressive	99.20%	13,174	106
Linear regression	98.34%	13059	221
Naive Bayes	93.79%	12455	825
Random Forest	89.85%	11,932	1,348
K-Neighbors	84.47%	11,218	2,062

4. Wizualizacja częstości występowania wyrazów za pomocą narzędzia WordCloud

FAŁSZYWE WIADOMOŚCI

PRAWDZIWE WIADOMOŚĆI



5. Wnioski końcowe

Zdaje się, że klasyfikacja wiadomości względem ich rzetelności jest jak najbardziej możliwa i do tego bardzo skuteczna, ale nie należy wyciągać pochopnych wniosków. Taka klasyfikacja może stanowić co najwyżej narzędzie poboczne pozwalające np. zgłaszać wiadomości wymagające weryfikacji gdyż w wypadku bardziej jakościowo napisanych „fejków" wyniki mogłyby być znacznie gorsze. Dodatkowo tematyka wiadomości znacząco zmienia się z roku na roku co najpewniej czyniłoby te klasyfikatory nieefektywnymi wobec nowszych treści.

Odnośnie samych klasyfikatorów, najlepiej zdają się sobie radzić Linear SVC oraz Passive-Aggressive Classifier, ze względu na bardzo dobre wyniki zarówno dla nagłówków jak i treści. Bardzo duża skuteczność Decision Tree Classifier dla treści swiadczy najpewniej o jego przeuczeniu. Linear Regression Classifier najlepiej poradził sobie w wypadku samych nagłówków, ale w wypadku korzystania z treści model robi się zbyt skomplikowany do opisania za pomocą regresji liniowej.