TERM WEIGHTING FOR FEATURE EXTRACTION ON TWITTER : A COMPARISON BETWEEN BM25 AND TF-IDF

AMMAR ISMAEL KADHIM , 2019 INTERNATIONAL CONFERENCE ON ADVANCED SCIENCE AND ENGINEERING (ICOASE)

組員: M10909112 石家安 M10909120 樊驊 M10915050 林紹瑜 M10915095 薛宇翔

- Introduction
- Method
- Experiment & Result
- Conclusion

Introduction

INTRODUCTION

- Due to the large volume of tweets in diverse topics, the classification of tweets on Twitter.
- Feature extraction is a significant stage in the process of initial text classification.
- Term frequency-inverse document frequency (TF-IDF) is normally utilized to weight the keywords
- BM25 can be defined as the ranked function that it used to rank corresponding text documents with respect to their relation to the specific search query.
- This paper introduces a comparative between BM25 and TF-IDF techniques for feature extraction.
- Experimental results show that the TF-IDF performs better than BM25 according to the performance classification metrics.

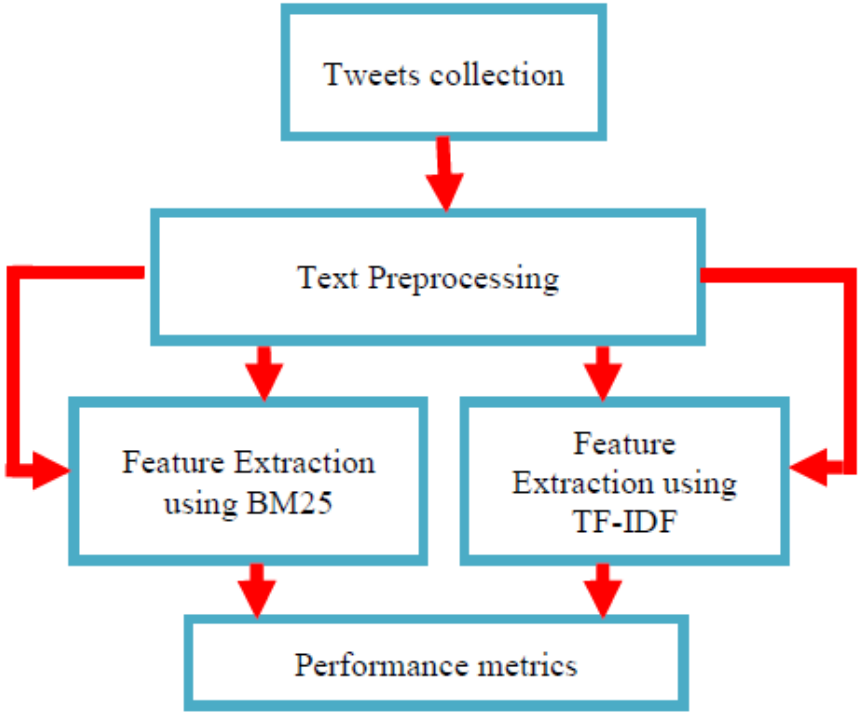
BACKGROUND

- Feature extraction is to transform a text document from any format into a list of features that can be easily processed by text classification techniques.

- Feature extraction is one of significant preprocessing techniques in data mining and text classification that computes features value in documents.

- This paper presents a comparative study of feature extraction techniques. Two techniques were evaluated BM25 and TF-IDF to weight the terms on Twitter.

PROPOSED APPROACH



- This paper comprises: Tweets collection using application programming interface (API), text preprocessing using tokenization, stop words and stemming
- Feature extraction using two different methods and comparison between TF-IDF and BM25 methods based on performance metrics

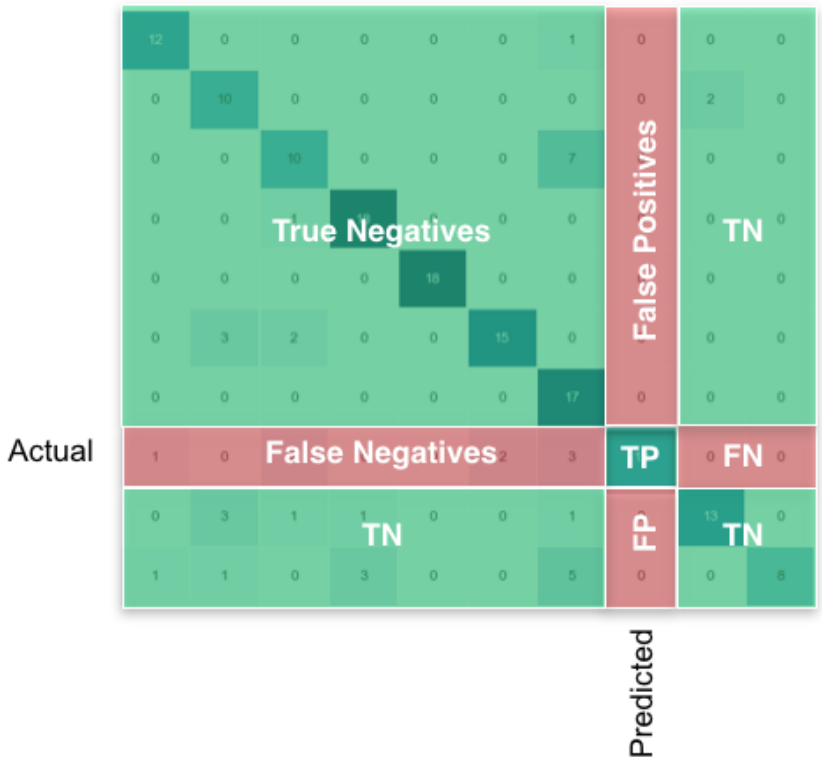
PERFORMANCE MATRIX(MULTI-CLASSIFICATION)

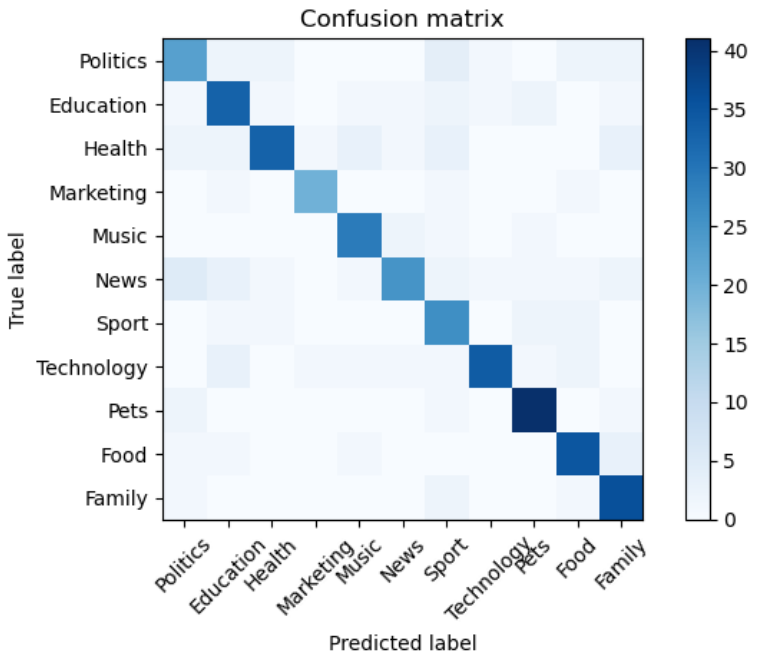
$$Acc_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}$$

$$P_i = \frac{TP_i}{TP_i + FP_i}$$

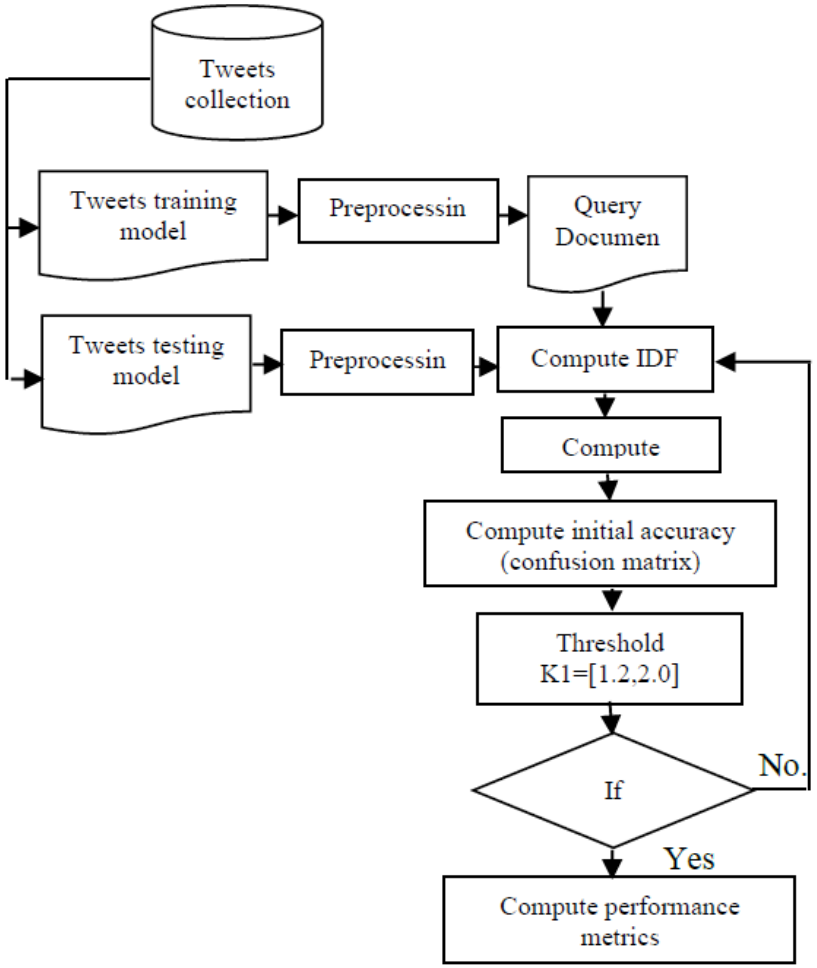
$$R_i = \frac{TP_i}{TP_i + FN_i}$$

$$F1 = \frac{2 \times P_i \times R_i}{(P_i + R_i)}$$



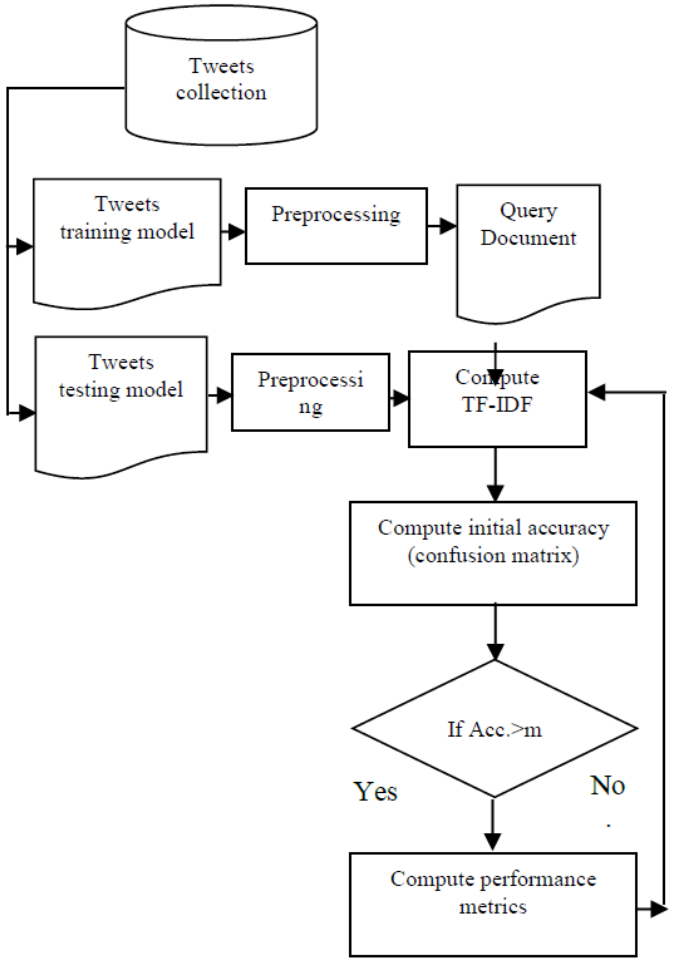


PROPOSED APPROACH - BM25

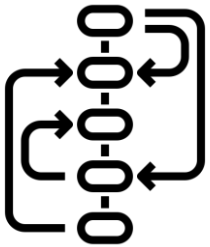


- Flowchart of BM25 method.

PROPOSED APPROACH - TF-IDF

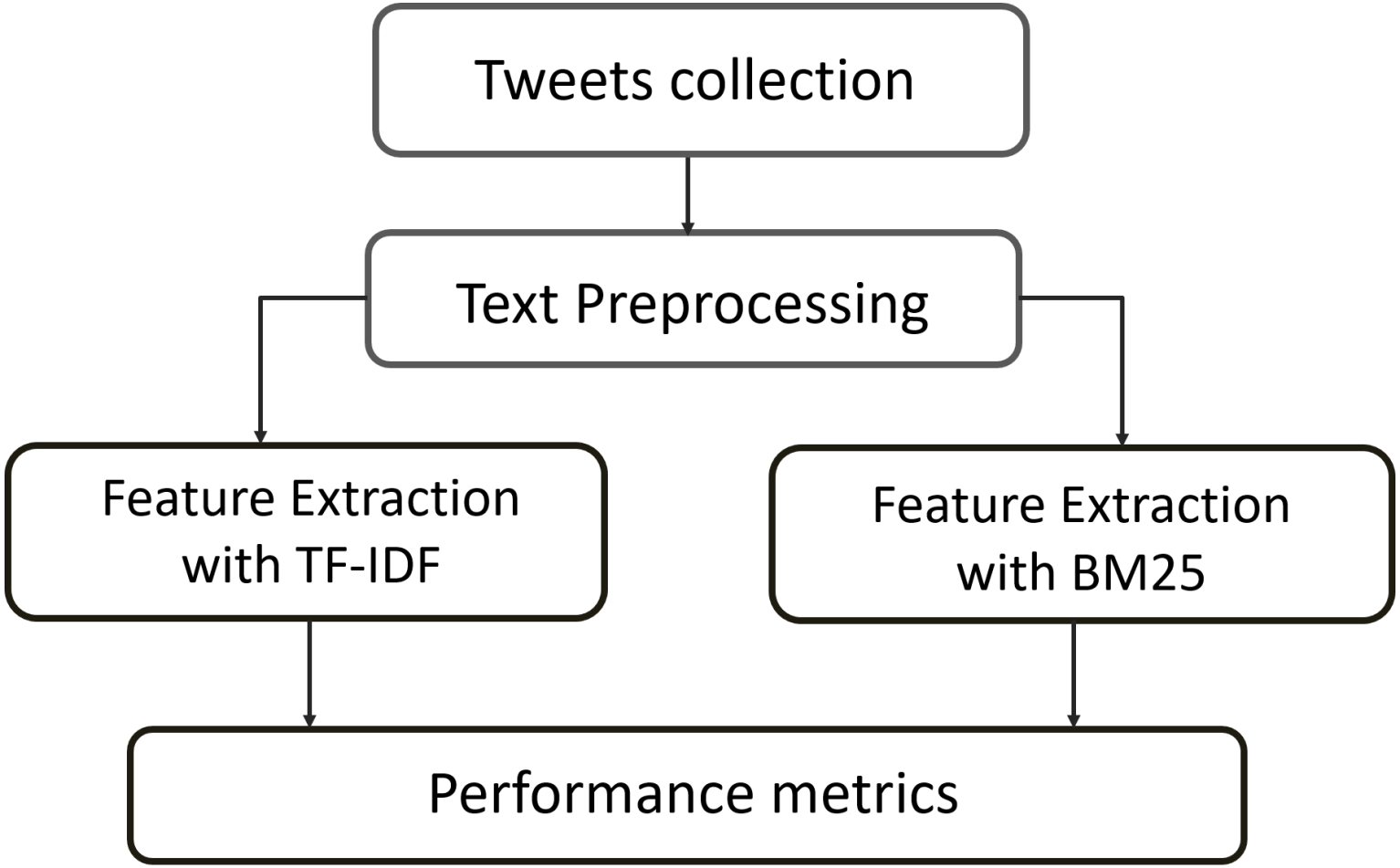


- A semantic step is very significant to select the proposed keywords that carried the same meaning of the tweet, and reject others tweets set.
- The whole procedure of feature extraction using TF-IDF as displayed in left.



Methodology

FLOW CHART


Tweets collection
Text Preprocessing
Feature Extraction with TF-IDF
Feature Extraction with BM25
Performance metrics

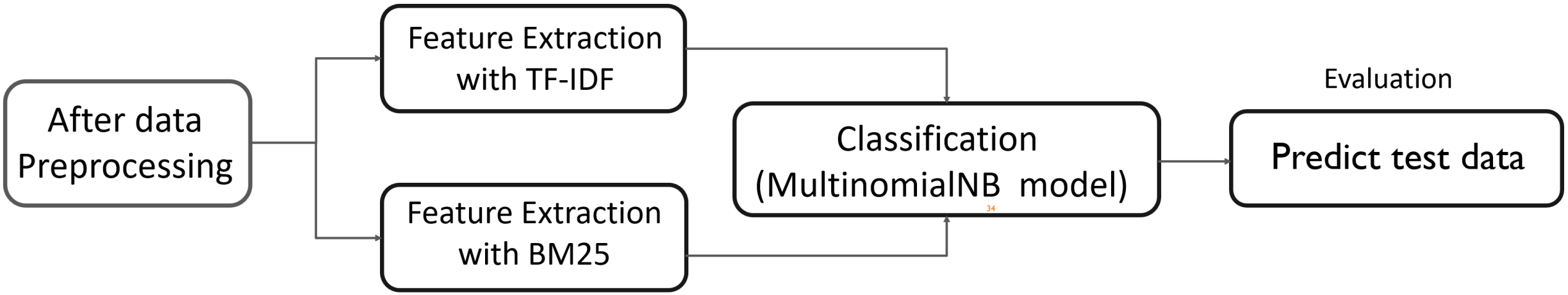
DATA COLLECTION & DATA PREPROCESSING

1. Using Tweepy package to request Twitter Streaming API
2. Remove noise text, like HTML, ‘RT’ and @people words
3. Text preprocessing using tokenization, lower case, stop words and snowball stemming, emoji to text…etc

OUR DATASET

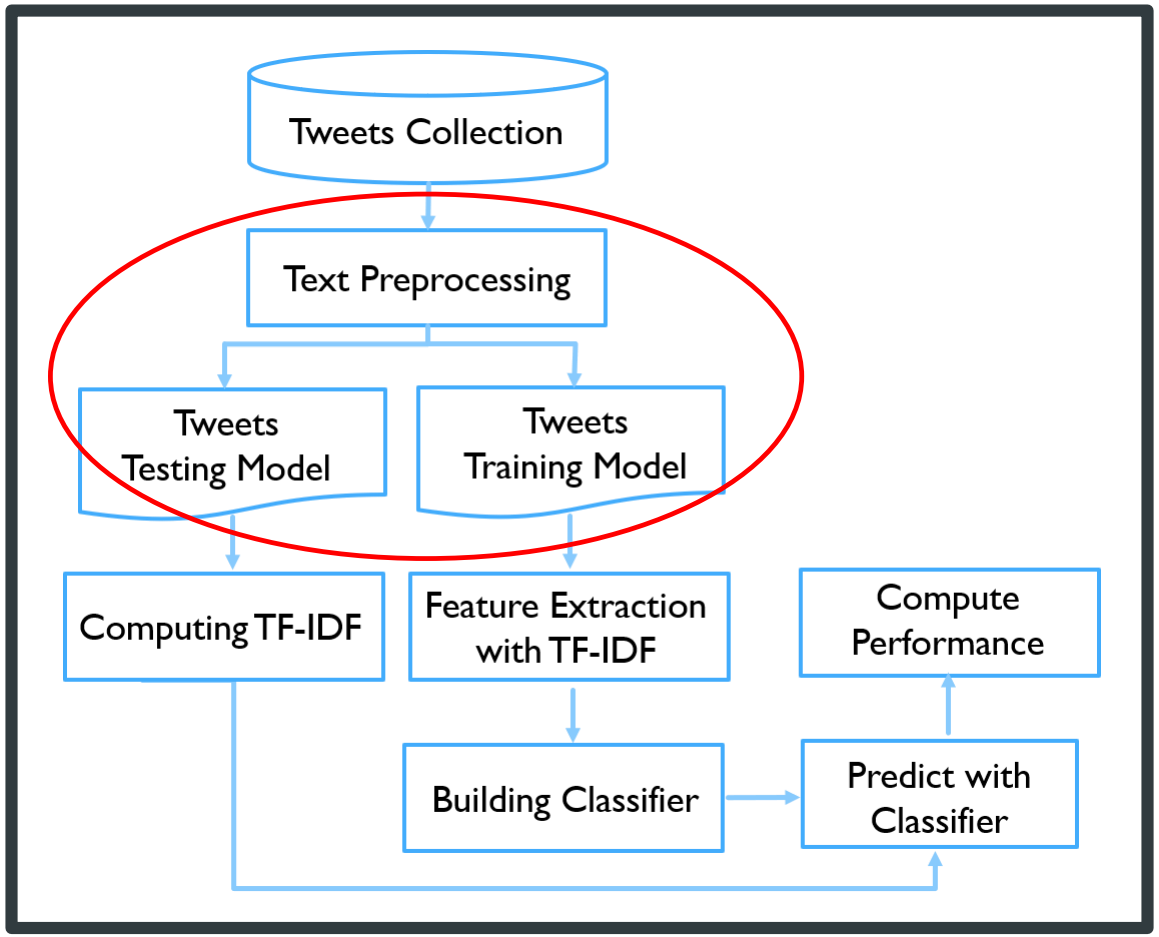
Class	Amount
Politics	200
Education	200
Health	200
Marketing	200
Music	200
News	200
Sport	200
Technology	200
Pets	200
Food	200
Family	200

TEXT CLASSIFICATION



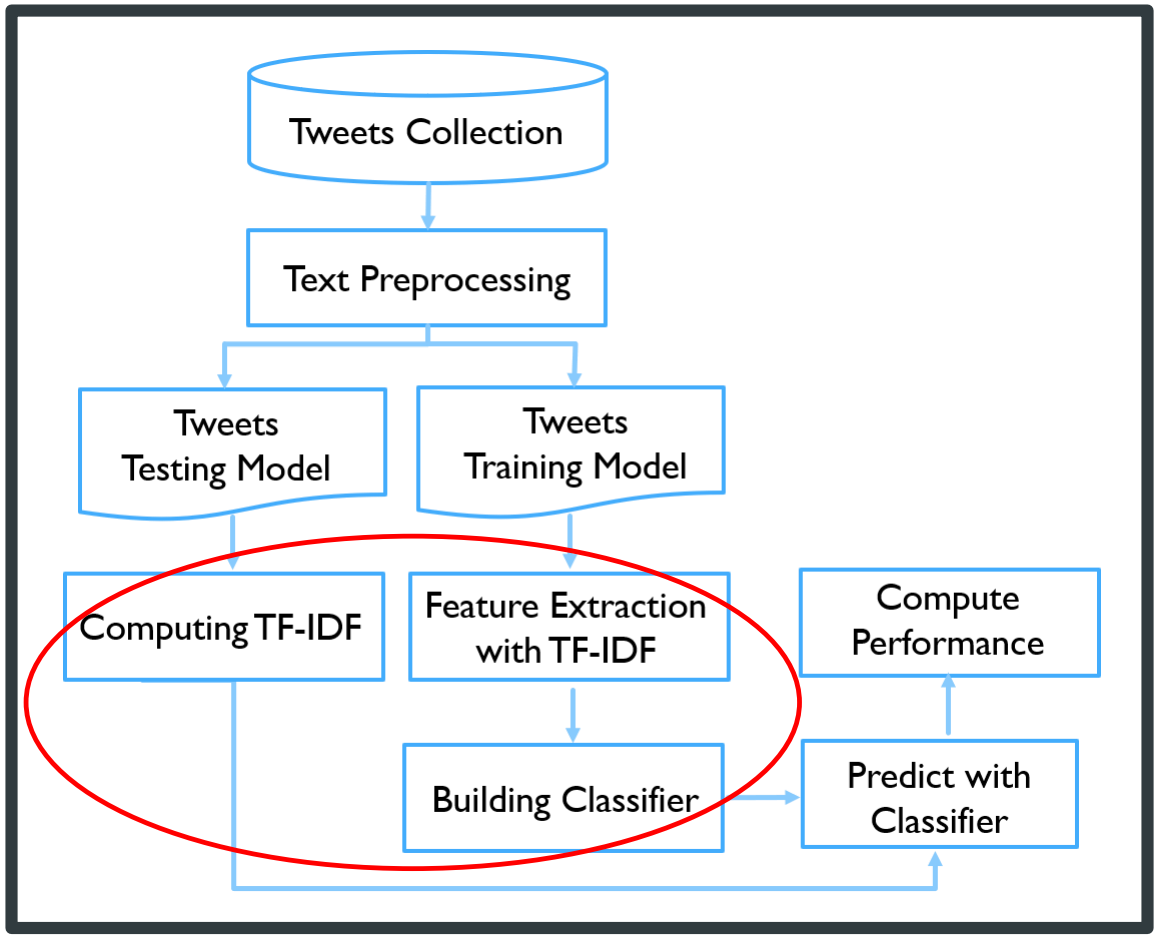
Experiments & Results

TF-IDF



- Preprocess collection
  - Emojitotext
  - remove_punctuation
  - Tokenize
  - remove_stopwords
  - word_lemmatizer
  - word_stemmer
- Split training and test data
  - Training data : 1760
  - Test data : 440

TF-IDF

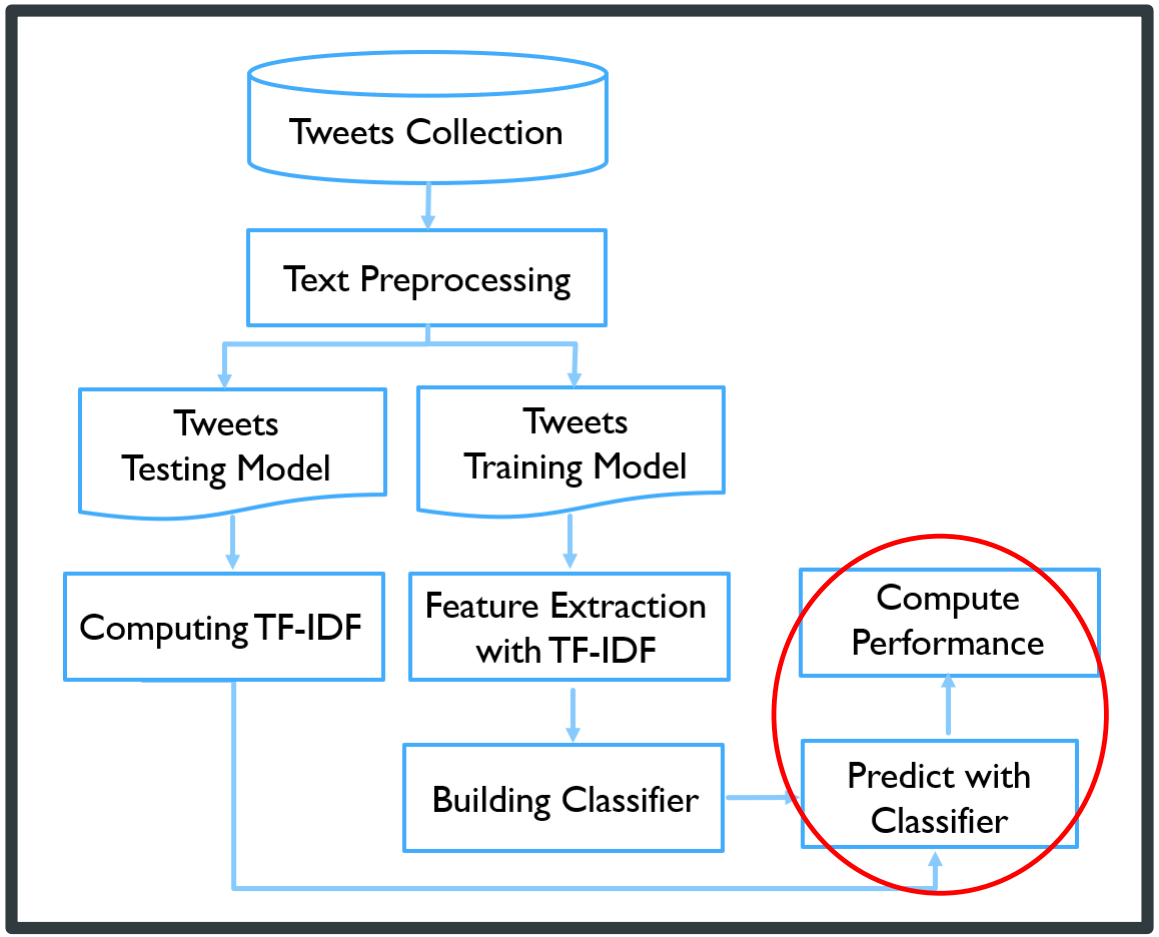


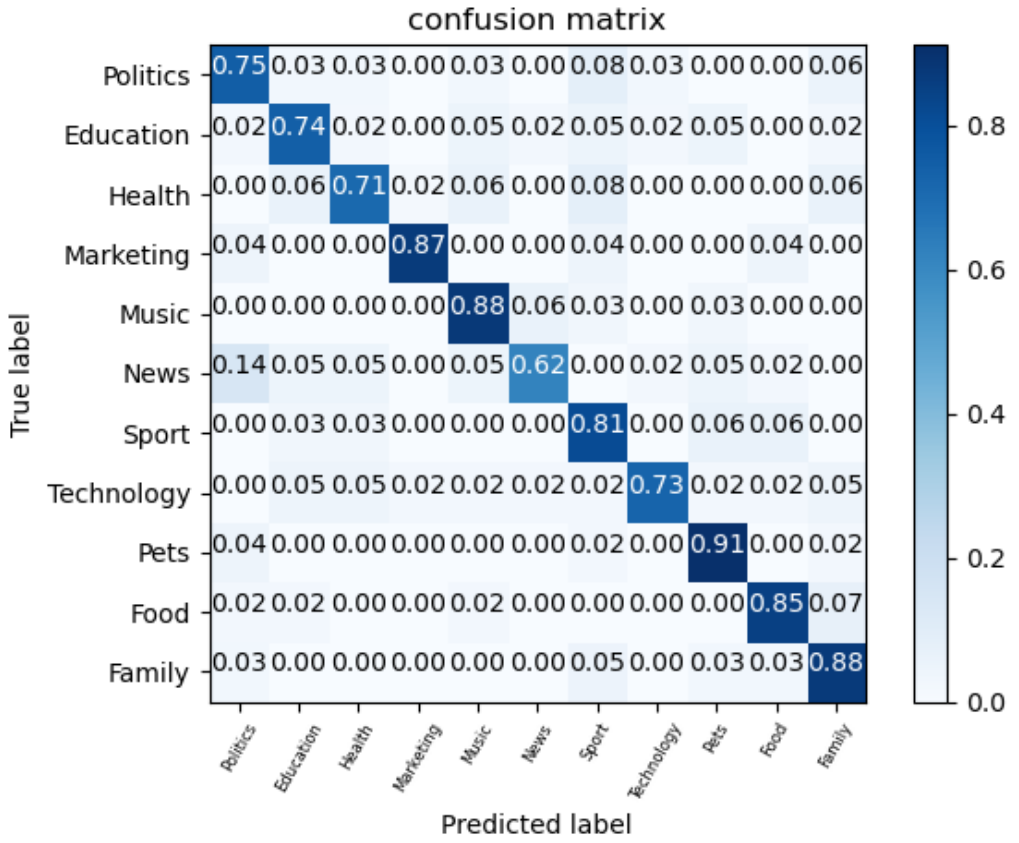
$$tf = 1 + \log(tf)$$

$$idf = \log\left(\frac{1+N}{1+n_i}\right) + \mathbf{1}$$

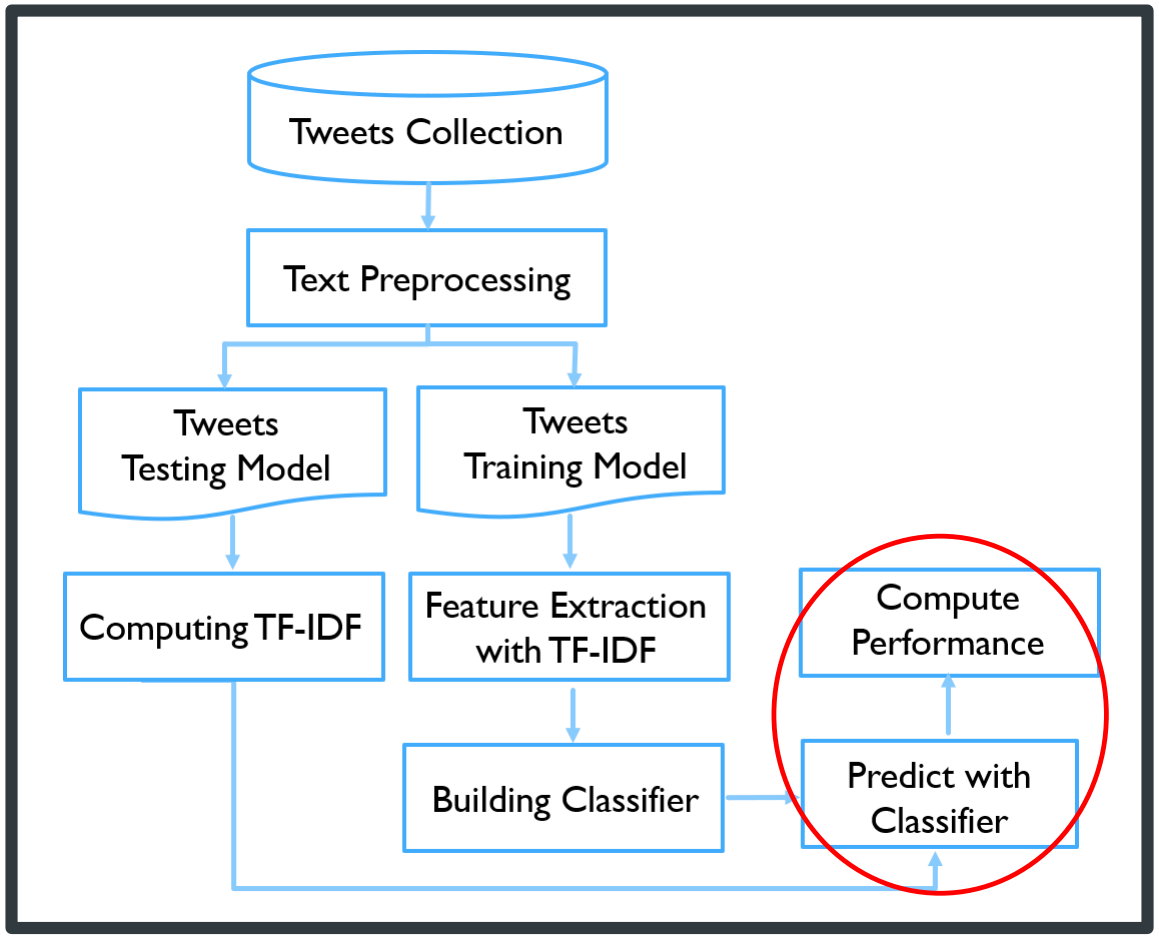
```
naive_bayes = MultinomialNB()
naive_bayes.fit(X_train_tfidf, y_train)
predictions = naive_bayes.predict(X_test_tfidf)
```

TF-IDF


Tweets Collection
Text Preprocessing
Tweets Testing Model
Tweets Training Model
Computing TF-IDF
Feature Extraction with TF-IDF
Compute Performance
Building Classifier
Predict with Classifier

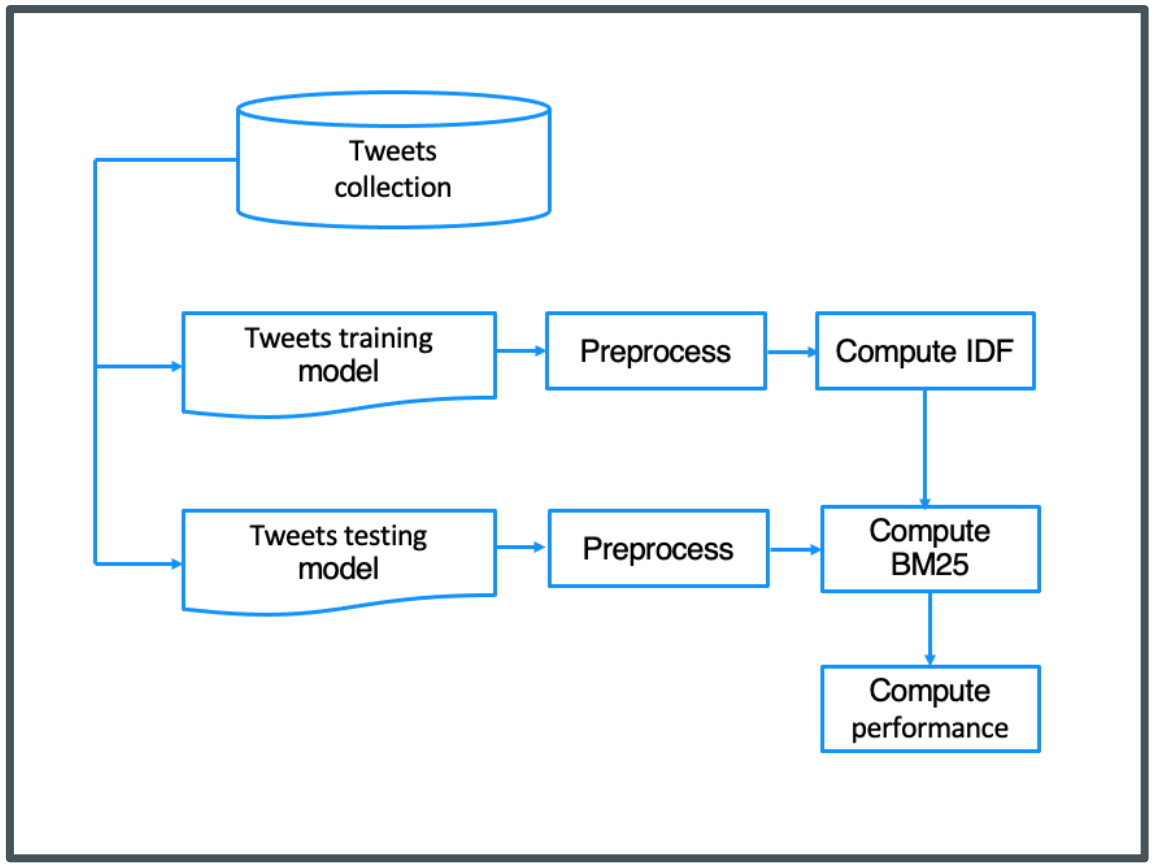

confusion matrix
Politics 0.75 0.03 0.03 0.00 0.03 0.00 0.08 0.03 0.00 0.00 0.06
Education 0.02 0.74 0.02 0.00 0.05 0.02 0.05 0.02 0.05 0.00 0.02
Health 0.00 0.06 0.71 0.02 0.06 0.00 0.08 0.00 0.00 0.00 0.06
Marketing 0.04 0.00 0.00 0.87 0.00 0.00 0.04 0.00 0.00 0.04 0.00
Music 0.00 0.00 0.00 0.00 0.88 0.06 0.03 0.00 0.03 0.00 0.00
News 0.14 0.05 0.05 0.00 0.05 0.62 0.00 0.02 0.05 0.02 0.00
Sport 0.00 0.03 0.03 0.00 0.00 0.00 0.81 0.00 0.06 0.06 0.00
Technology 0.00 0.05 0.05 0.02 0.02 0.02 0.02 0.73 0.02 0.02 0.05
Pets 0.04 0.00 0.00 0.00 0.00 0.00 0.02 0.00 0.91 0.00 0.02
Food 0.02 0.02 0.00 0.00 0.02 0.00 0.00 0.00 0.00 0.85 0.07
Family 0.03 0.00 0.00 0.00 0.00 0.00 0.05 0.00 0.03 0.03 0.88
True label
Politics Education Health Marketing Music News Sport Technology Pets Food Family
Predicted label
0.8
0.6
0.4
0.2
0.0

TF-IDF



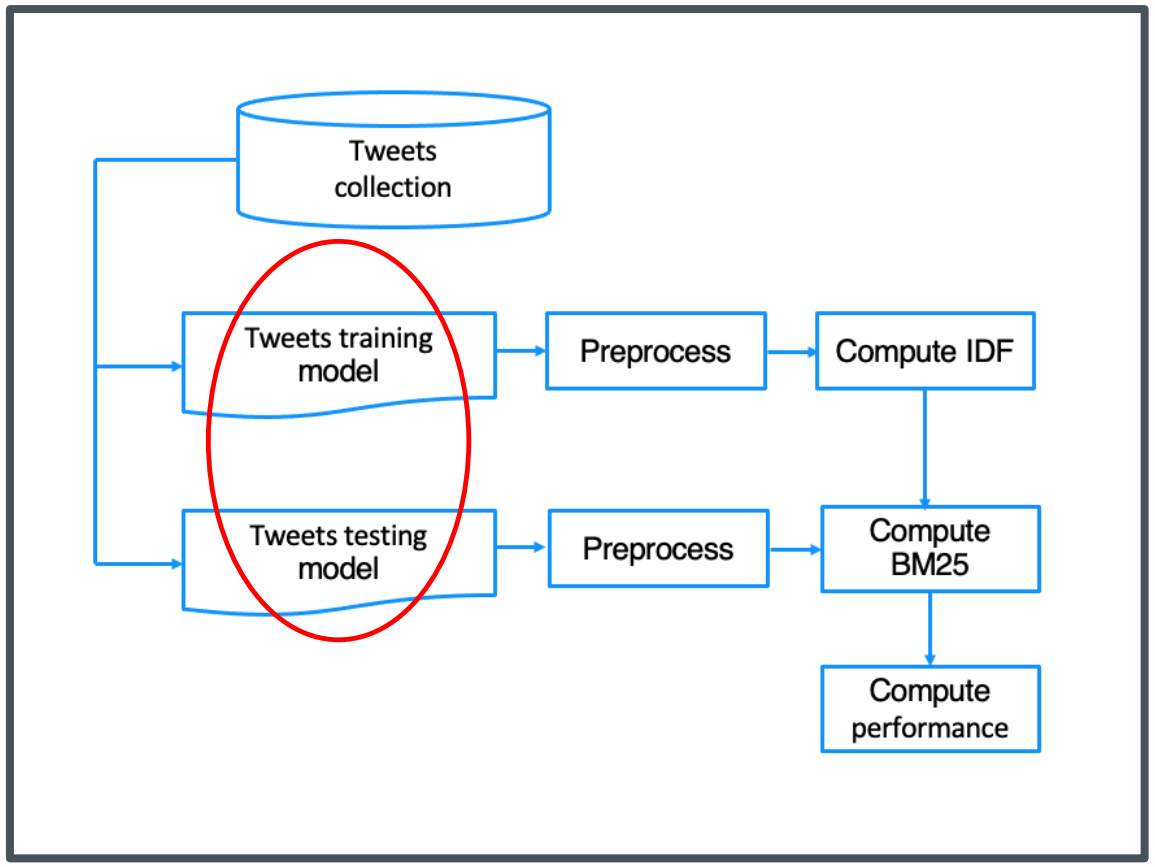
	accuracy	precision	recall	F1
Politics	0.95	0.71	0.75	0.73
Education	0.96	0.76	0.74	0.75
Health	0.96	0.80	0.71	0.75
Marketing	0.98	0.95	0.87	0.91
Music	0.97	0.79	0.88	0.83
News	0.96	0.85	0.62	0.72
Sport	0.95	0.68	0.81	0.74
Technology	0.97	0.91	0.73	0.81
Pets	0.97	0.80	0.91	0.85
Food	0.97	0.83	0.85	0.84
Family	0.96	0.76	0.88	0.81

BM25



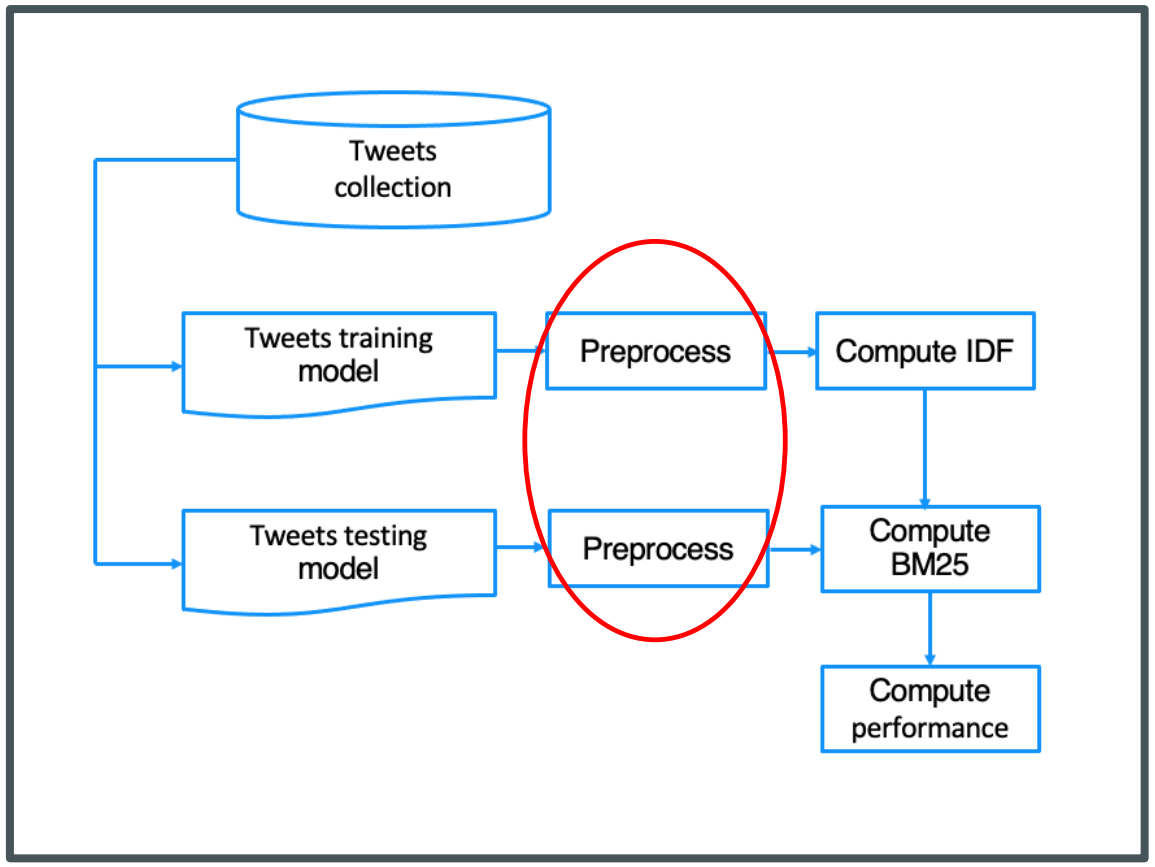
- Step 1: Preprocess data
- Step 2: Compute IDF by training data (global info.)
- Step 3: Transform data by BM25
- Step 4: Fit Naive Bayes classifier by training set
- Step 5: Predict test data set result

BM25



- Preprocess collection and split
  - Emojitotext
  - remove_punctuation
  - Tokenize
  - remove_stopwords
  - word_lemmatizer
  - word_stemmer
- Split training and test data
  - Training data : 1760
  - Test data : 440

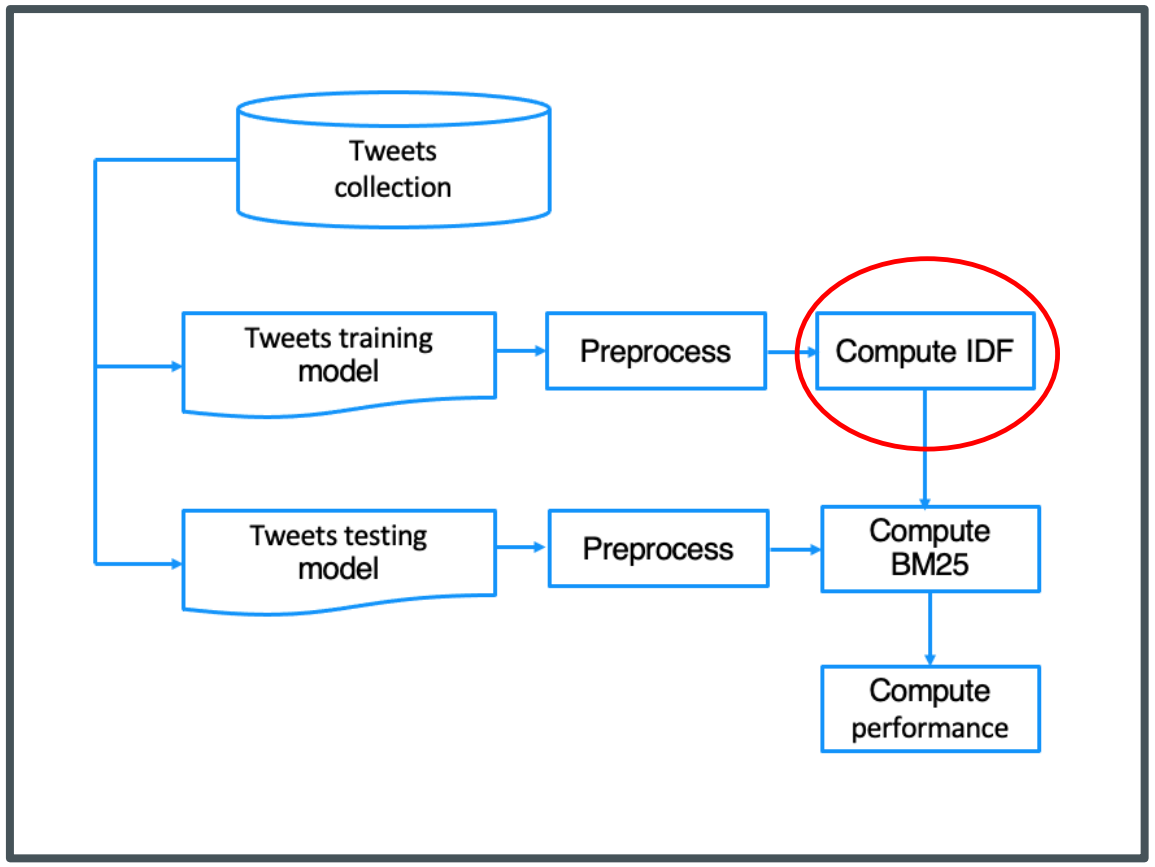
BM25




```
tfidf_vectorizer = CountVectorizer(
    min_df=1, max_features=720,
    strip_accents='unicode', analyzer='word',
    ngram_range=(1, 1), lowercase=True,
    input='content', encoding='utf-8',
    decode_error='strict', preprocessor=None,
    tokenizer=None, stop_words=None,
    token_pattern=r"\w+", max_df=1.0,
    vocabulary=None, binary=False, dtype=np.float64
)
```

- CountVectorizer
- min_df=1,
  max_features=720,
  strip_accents='unicode',
  analyzer='word',
  ngram_range=(1, 1),
  lowercase=True,
  input='content', encoding='utf-8',
  decode_error='strict',
  preprocessor=None,
  tokenizer=None,
  stop_words=None,
  token_pattern=r"\w+",
  max_df=1.0,
  vocabulary=None,
  binary=False,
  dtype=np.float64

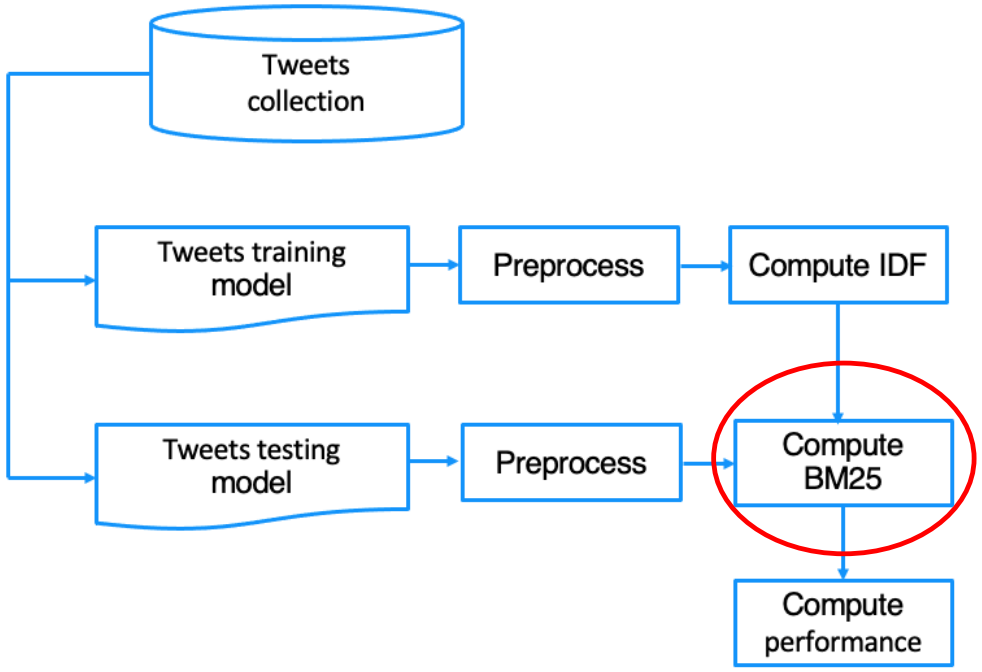
BM25



$$log \frac{N - n_i}{n_i}$$

```
bm25idf = np.log((n_samples - df) / df)
```

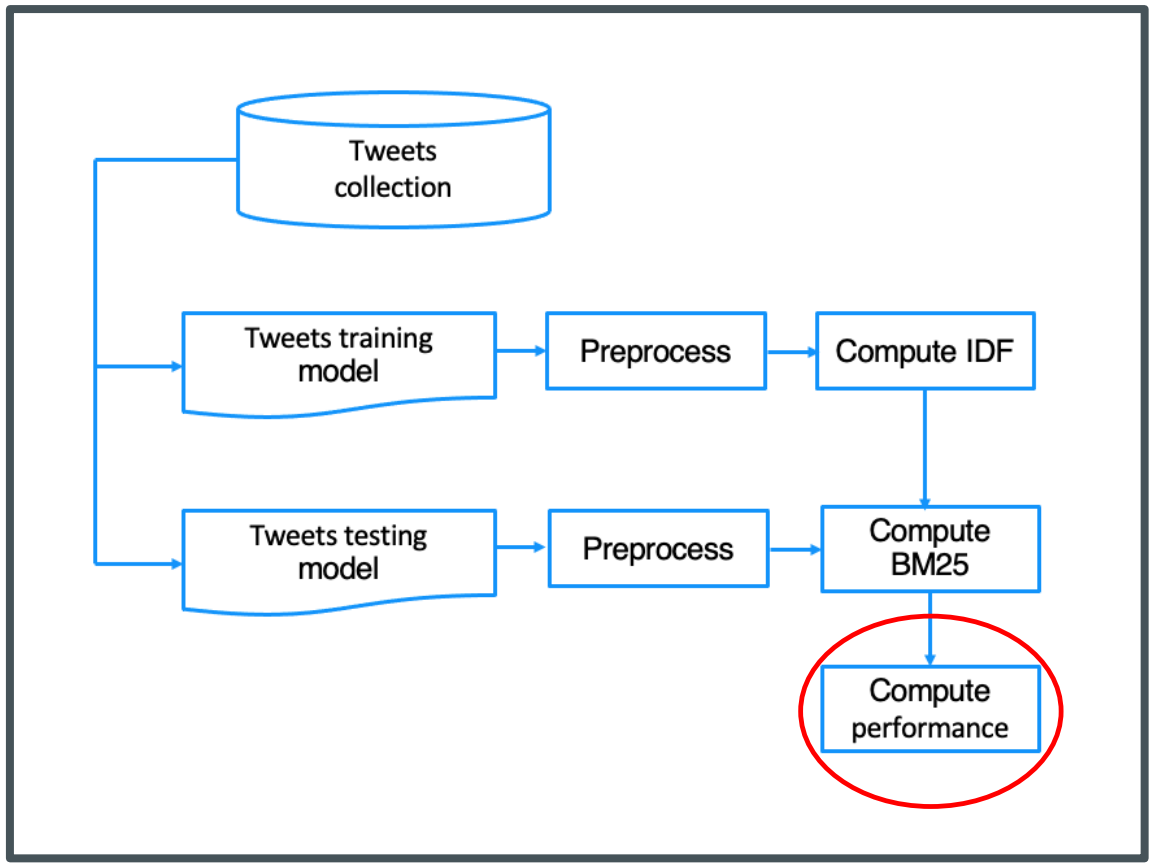
BM25

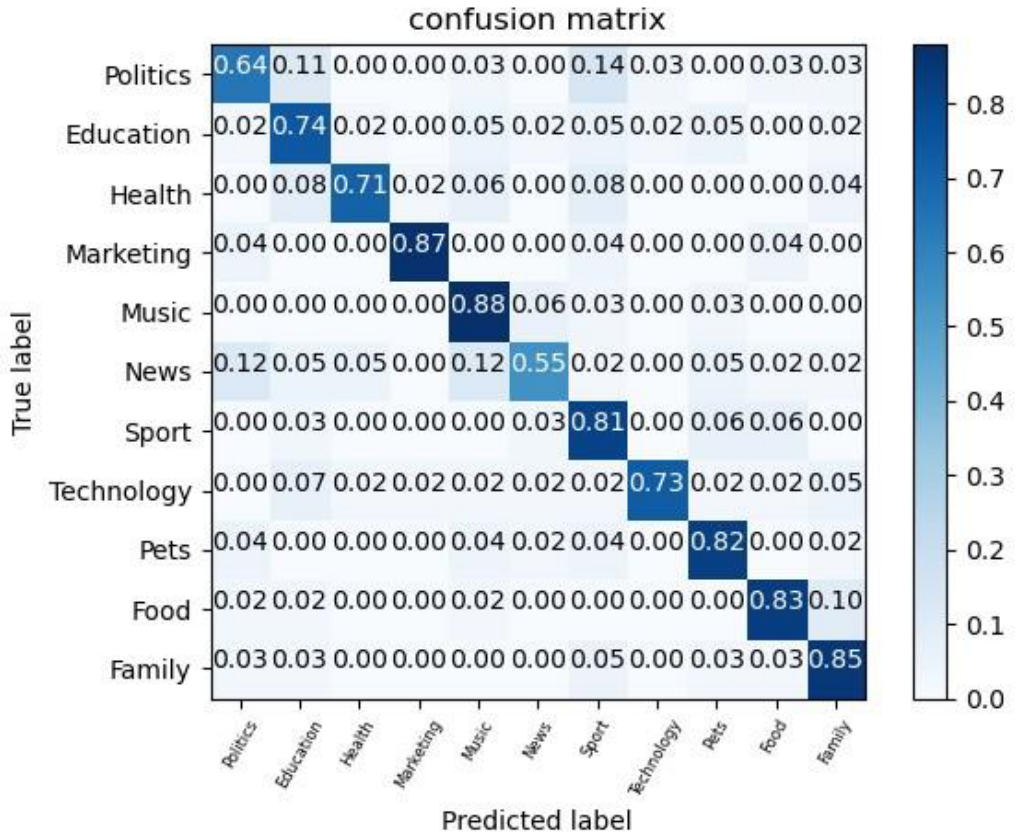


$$BM25 = score(D,Q) = \sum_{i=1}^{n} IDF(q_i) \bullet \frac{f(q_i,D) \bullet (k_1+1)}{f(q_i,D) + k_1 \bullet (1 - b + b \bullet \frac{|D|}{avgdl})}$$

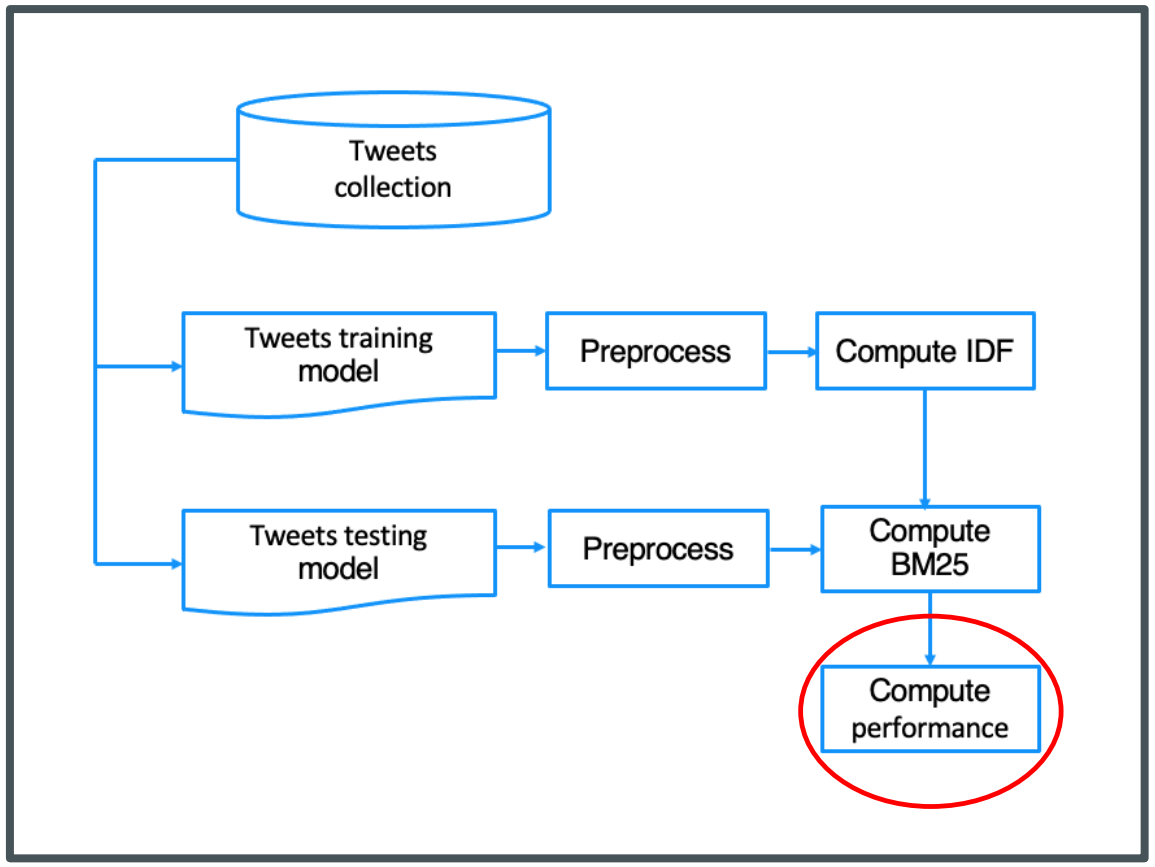
```
D = (X.sum(1) / np.average(X.sum(1))).reshape((n_samples, 1))
D = ((1 - self.b) + self.b * D) * self.k
D_X = _add_sparse_column(X, D)
X.data = np.divide(X.data * (self.k + 1), D_X.data, X.data)
X = np.dot(X, self._idf_diag)
```

BM25


Tweets collection
Tweets training model
Preprocess
Compute IDF
Tweets testing model
Preprocess
Compute BM25
Compute performance


confusion matrix
Politics 0.64 0.11 0.00 0.00 0.03 0.00 0.14 0.03 0.00 0.03 0.03
Education 0.02 0.74 0.02 0.00 0.05 0.02 0.05 0.02 0.05 0.00 0.02
Health 0.00 0.08 0.71 0.02 0.06 0.00 0.08 0.00 0.00 0.00 0.04
Marketing 0.04 0.00 0.00 0.87 0.00 0.00 0.04 0.00 0.00 0.04 0.00
Music 0.00 0.00 0.00 0.00 0.88 0.06 0.03 0.00 0.03 0.00 0.00
News 0.12 0.05 0.05 0.00 0.12 0.55 0.02 0.00 0.05 0.02 0.02
Sport 0.00 0.03 0.00 0.00 0.00 0.03 0.81 0.00 0.06 0.06 0.00
Technology 0.00 0.07 0.02 0.02 0.02 0.02 0.02 0.73 0.02 0.02 0.05
Pets 0.04 0.00 0.00 0.00 0.04 0.02 0.04 0.00 0.82 0.00 0.02
Food 0.02 0.02 0.00 0.00 0.02 0.00 0.00 0.00 0.00 0.83 0.10
Family 0.03 0.03 0.00 0.00 0.00 0.00 0.05 0.00 0.03 0.03 0.85
True label
Politics Education Health Marketing Music News Sport Technology Pets Food Family
Predicted label
0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0

BM25



	accuracy	precision	recall	F1
Politics	0.94	0.70	0.64	0.67
Education	0.94	0.66	0.74	0.70
Health	0.96	0.88	0.71	0.79
Marketing	0.98	0.95	0.87	0.91
Music	0.96	0.72	0.88	0.79
News	0.94	0.77	0.55	0.64
Sport	0.94	0.63	0.81	0.71
Technology	0.97	0.93	0.73	0.82
Pets	0.96	0.78	0.82	0.80
Food	0.97	0.80	0.83	0.82
Family	0.96	0.75	0.85	0.80

Conclusion

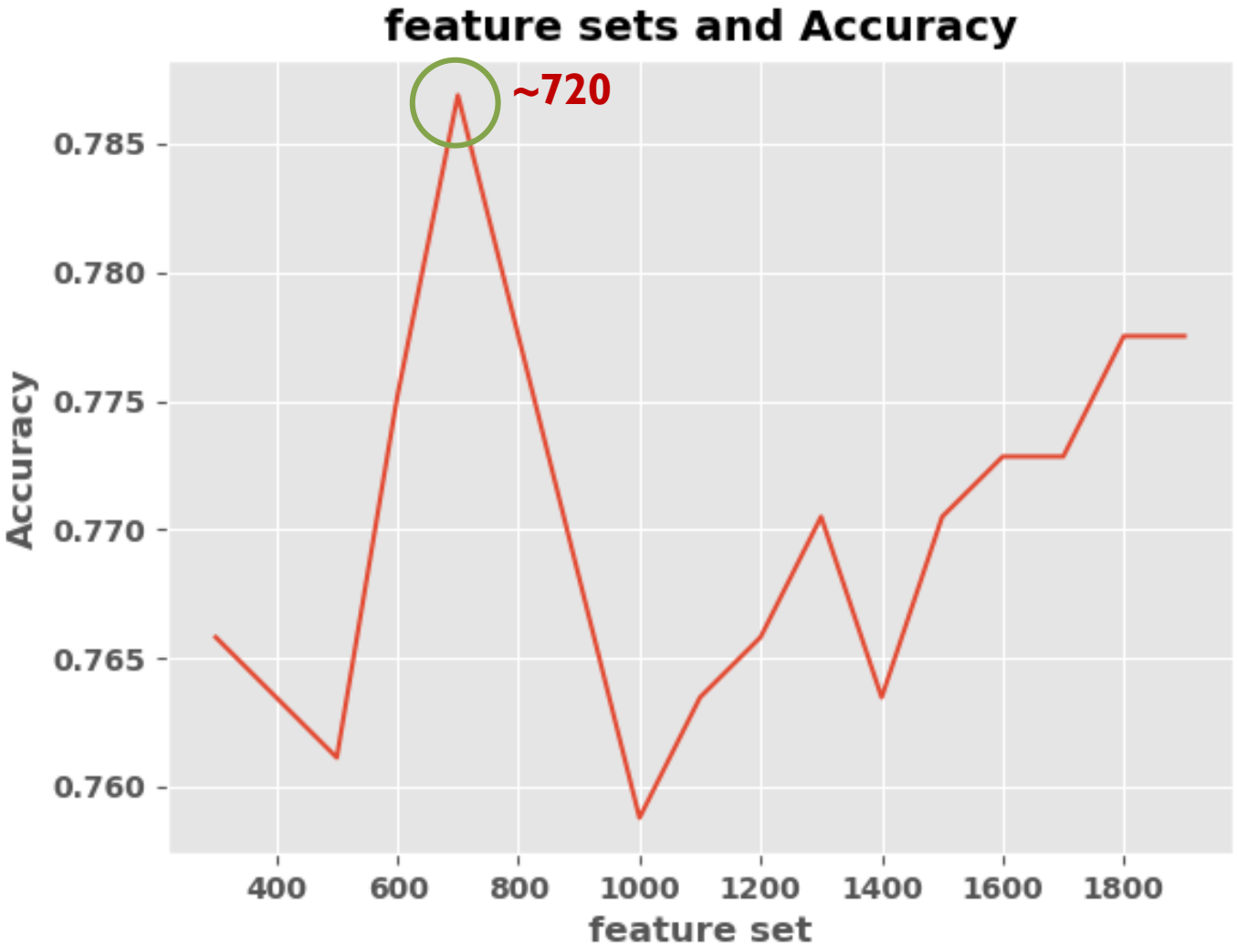

feature sets and Accuracy
~720
0.785
0.780
0.775
0.770
0.765
0.760
Accuracy
400
600
800
1000
1200
1400
1600
1800
feature set

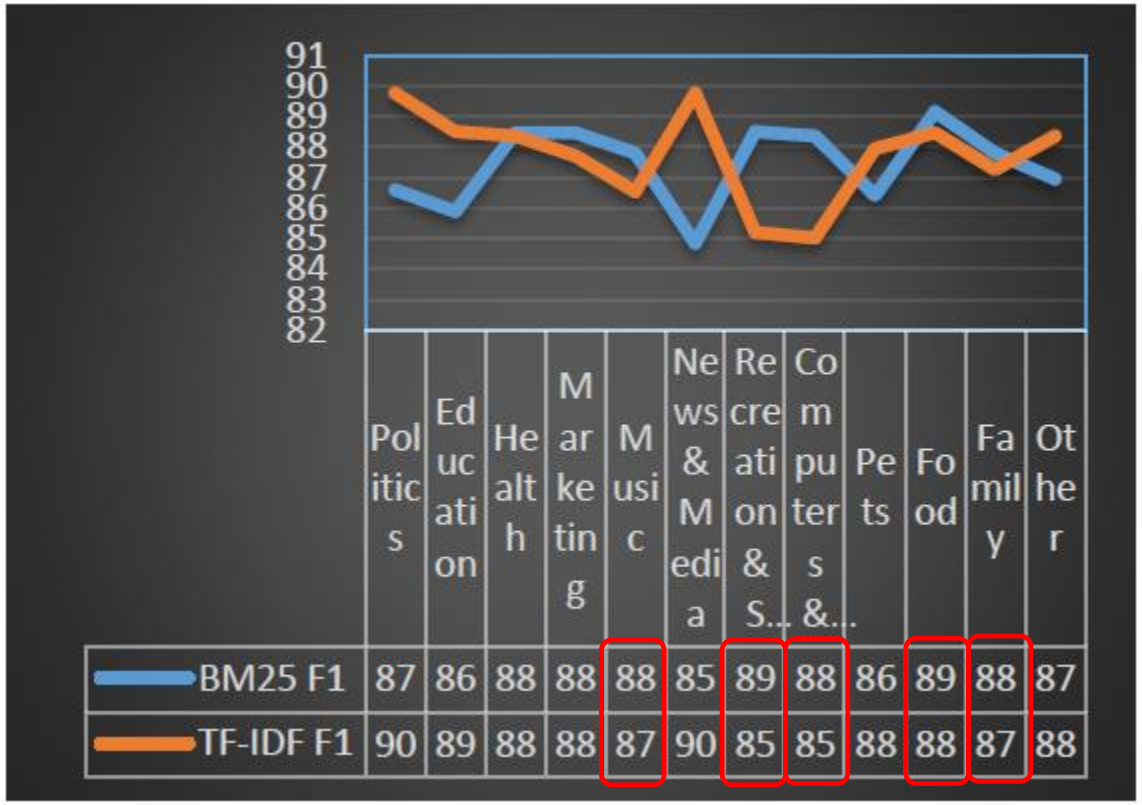
TABLE II. PERFORMANCE CLASSIFICATION USING TF-IDF METHOD.

Category	TF-IDF			
	Acc.	P	R	F1
Politics	82.00	90.80	88.76	89.77
Education	80.00	88.51	88.51	88.51
Health	80.00	86.36	90.48	88.37
Marketing	79.00	87.21	88.24	87.72
Music	77.00	85.06	88.10	86.55
News & Media	82.00	90.80	88.76	89.77
Recreation & Sports	75.00	83.72	86.75	85.21
Computers & Technology	75.00	85.54	84.52	85.03
Pets	80.00	85.88	90.12	87.95
Food	81.00	90.12	86.90	88.48
Family	79.00	87.80	86.75	87.27
Other	80.00	90.48	86.36	88.37

TABLE I. PERFORMANCE CLASSIFICATION USING BM 25 METHOD.

Category	BM25			
	Acc.	*P*	*R*	*F1*
Politics	78.00	86.59	86.59	86.59
Education	77.00	86.42	85.37	85.89
Health	81.00	87.95	89.02	88.48
Marketing	81.00	86.90	90.12	88.48
Music	80.00	87.80	87.80	87.80
News & Media	75.00	85.37	84.37	84.85
Recreation & Sports	80.00	88.51	88.51	88.51
Computers & Technology	80.00	86.36	90.48	88.37
Pets	78.00	86.42	86.42	86.42
Food	82.00	89.16	89.16	89.16
Family	80.00	86.75	88.89	87.80
Other	79.00	86.42	87.50	86.96

▼ Comparison between BM25 and TF-IDF according to F1-measure.



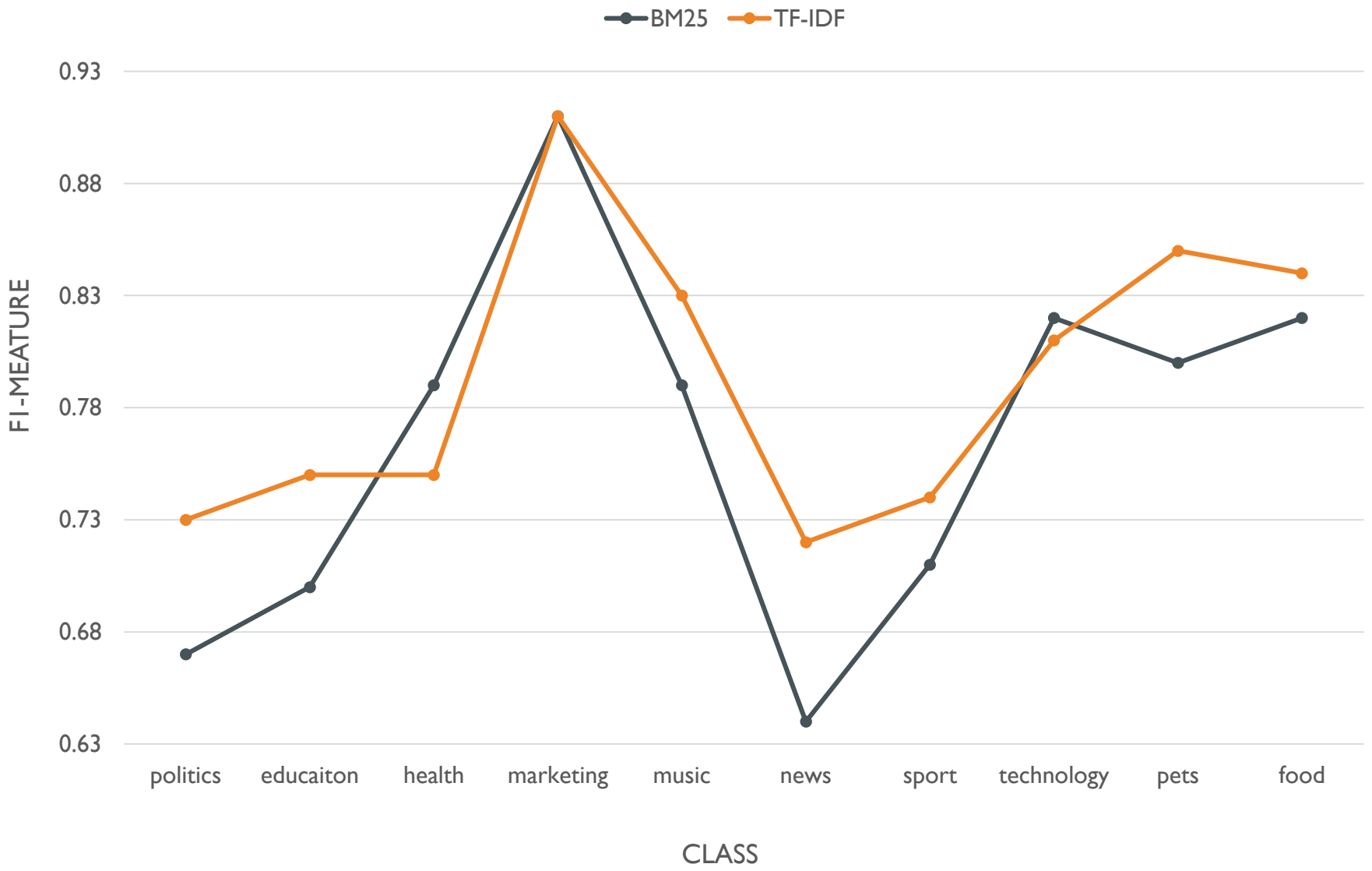
▼TF-IDF

	accuracy	precision	recall	F1
Politics	0.95	0.71	0.75	0.73
Education	0.96	0.76	0.74	0.75
Health	0.96	0.80	0.71	0.75
Marketing	0.98	0.95	0.87	0.91
Music	0.97	0.79	0.88	0.83
News	0.96	0.85	0.62	0.72
Sport	0.95	0.68	0.81	0.74
Technology	0.97	0.91	0.73	0.81
Pets	0.97	0.80	0.91	0.85
Food	0.97	0.83	0.85	0.84
Family	0.96	0.76	0.88	0.81

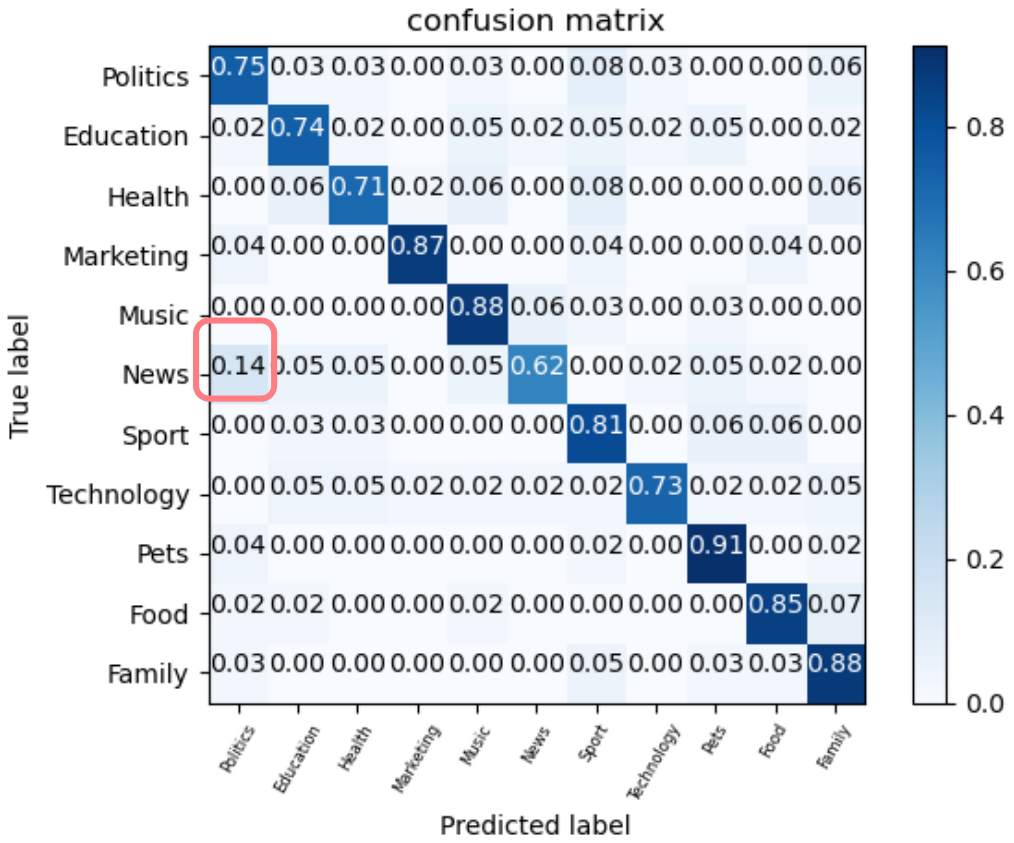
▼BM25

	accuracy	precision	recall	F1
Politics	0.94	0.70	0.64	0.67
Education	0.94	0.66	0.74	0.70
Health	0.96	0.88	0.71	0.79
Marketing	0.98	0.95	0.87	0.91
Music	0.96	0.72	0.88	0.79
News	0.94	0.77	0.55	0.64
Sport	0.94	0.63	0.81	0.71
Technology	0.97	0.93	0.73	0.82
Pets	0.96	0.78	0.82	0.80
Food	0.97	0.80	0.83	0.82
Family	0.96	0.75	0.85	0.80

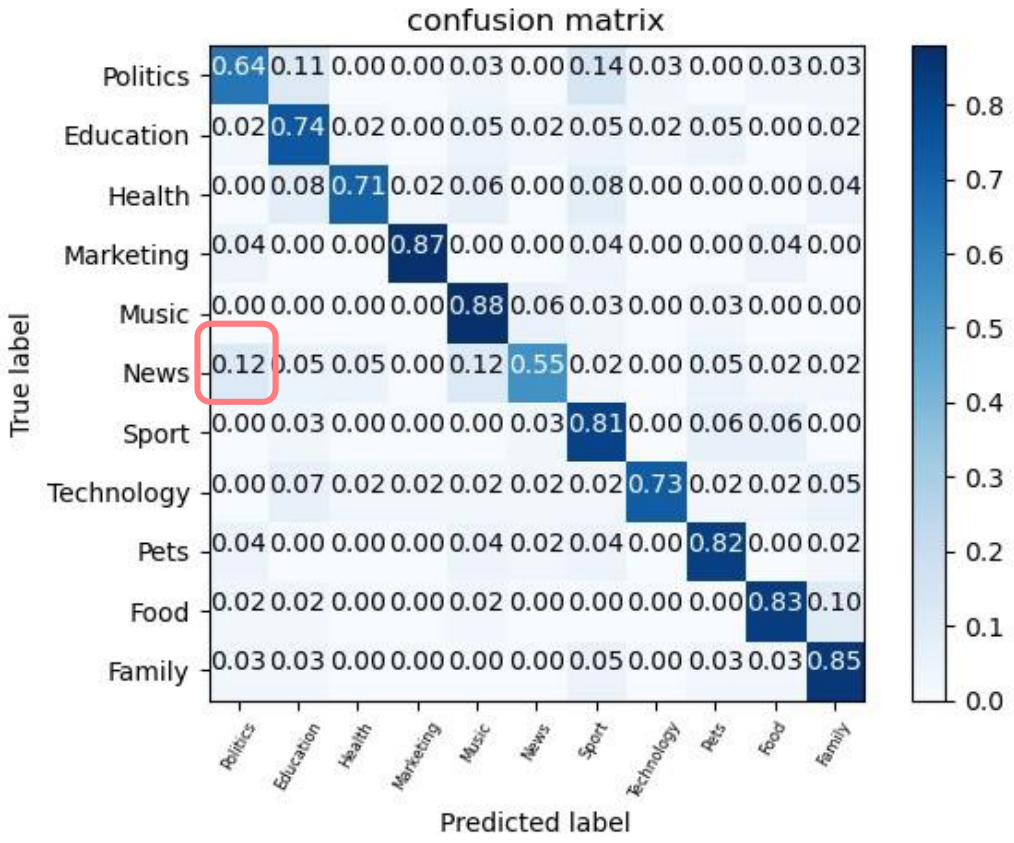
Comparision between BM25 and TF-IDF according to F1-measure



▼TF-IDF


confusion matrix
Politics 0.75 0.03 0.03 0.00 0.03 0.00 0.08 0.03 0.00 0.00 0.06
Education 0.02 0.74 0.02 0.00 0.05 0.02 0.05 0.02 0.05 0.00 0.02
Health 0.00 0.06 0.71 0.02 0.06 0.00 0.08 0.00 0.00 0.00 0.06
Marketing 0.04 0.00 0.00 0.87 0.00 0.00 0.04 0.00 0.00 0.04 0.00
Music 0.00 0.00 0.00 0.00 0.88 0.06 0.03 0.00 0.03 0.00 0.00
News 0.14 0.05 0.05 0.00 0.05 0.62 0.00 0.02 0.05 0.02 0.00
Sport 0.00 0.03 0.03 0.00 0.00 0.00 0.81 0.00 0.06 0.06 0.00
Technology 0.00 0.05 0.05 0.02 0.02 0.02 0.02 0.73 0.02 0.02 0.05
Pets 0.04 0.00 0.00 0.00 0.00 0.00 0.02 0.00 0.91 0.00 0.02
Food 0.02 0.02 0.00 0.00 0.02 0.00 0.00 0.00 0.00 0.85 0.07
Family 0.03 0.00 0.00 0.00 0.00 0.00 0.05 0.00 0.03 0.03 0.88
Politics Education Health Marketing Music News Sport Technology Pets Food Family
True label
Predicted label
0.8 0.6 0.4 0.2 0.0

▼BM25


confusion matrix
Politics 0.64 0.11 0.00 0.00 0.03 0.00 0.14 0.03 0.00 0.03 0.03
Education 0.02 0.74 0.02 0.00 0.05 0.02 0.05 0.02 0.05 0.00 0.02
Health 0.00 0.08 0.71 0.02 0.06 0.00 0.08 0.00 0.00 0.00 0.04
Marketing 0.04 0.00 0.00 0.87 0.00 0.00 0.04 0.00 0.00 0.04 0.00
Music 0.00 0.00 0.00 0.00 0.88 0.06 0.03 0.00 0.03 0.00 0.00
News 0.12 0.05 0.05 0.00 0.12 0.55 0.02 0.00 0.05 0.02 0.02
Sport 0.00 0.03 0.00 0.00 0.00 0.03 0.81 0.00 0.06 0.06 0.00
Technology 0.00 0.07 0.02 0.02 0.02 0.02 0.02 0.73 0.02 0.02 0.05
Pets 0.04 0.00 0.00 0.00 0.04 0.02 0.04 0.00 0.82 0.00 0.02
Food 0.02 0.02 0.00 0.00 0.02 0.00 0.00 0.00 0.00 0.83 0.10
Family 0.03 0.03 0.00 0.00 0.00 0.00 0.05 0.00 0.03 0.03 0.85
Politics Education Health Marketing Music News Sport Technology Pets Food Family
True label
Predicted label
0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0

REFERENCE

1. Twitter API tutorial : https://developer.twitter.com/en/docs
2. Emoji to text : https://pypi.org/project/demoji/
3. Stop words : https://github.com/stopwords-iso/stopwords-en
4. Bayesian Classifier : https://pyecontech.com/2020/03/06/python%E5%AF%A6%E4%BD%9C-%E8%B2%9D%E6%B0%8F%E5%88%86%E9%A1%9E%E5%99%A8-bayesian-classifier/

謝謝大家