Assignment 1: Text Categorization

Group 1
Bangchao Xie (s3537145), Shupei Li (s3430863)

1 Methodology

1.1 Classifiers

We choose naïve bayes (NB), support vector machine (SVM), and random forest (RF) to categorize text. Tutorial in sklearn suggests that the multinomial variant of naïve bayes and a linear support vector machine are suitable for this task. Therefore, we use `MultinomialNB` and `SGDClassifier` APIs in sklearn. Random forest is an ensemble machine learning method based on multiple decision tree models. We use the `RandomForestClassifier` implementation in sklearn. Table 1 summarizes key hyperparameters in three classifiers. In experiments, hyperparameters are optimized with grid search.

Table 1: Parameters in Three Classifiers

Classifier	**Parameter**	**Meaning**
Naïve Bayes	alpha	Additive smoothing parameter.
Support Vector Machine	alpha	Constant that multiplies the regularization term.
Random Forest	n_estimators	The number of trees in the forest.

1.2 Features

Counts, tf, and tf-idf are adopted as features. According to instructions in sklearn tutorial, counts feature is extracted by `CountVectorizer` API, while tf and tf-idf are generated by `TfidfTransformer` API. In task 4, we set hyperparameters in `CountVectorizer` to different values, as described in Table 2.

Table 2: Parameters in `CountVectorizer`

Task	**Parameter**	**Values***
4.a	lowercase	True, False
4.b	stop_words	None, "english"
4.c	analyzer	"word", "char"
	ngram_range	(1, 1), (1, 2)
4.d	max_features**	None, 115000

* The first value is the default value in `CountVectorizer`.
** max_features is set to 0.9 quantile of corpus size.

2 Results

Precision, recall, and F1 are selected as metrics to evaluate the model performance.

2.1 Task 1 to 3

Table 3 reports our experiment results in Task 1, 2, and 3.

Table 3: Results of Task 1 to 3

Classifier	Feature	Param.(clf)*	Precision	Recall	F1
NB	count	0.01	0.8082	0.7999	0.7852
	tf	**0.0001**	**0.8244**	**0.8204**	**0.8204**
	tf-idf	**0.0001**	**0.8244**	**0.8204**	0.**8204**
SVM	count	0.1	0.7841	0.7789	0.7772
	tf	0.0001	0.8102	0.7966	0.7961
	tf-idf	0.0001	0.8060	0.7995	0.7988
RF	count	200	0.7913	0.7696	0.7687
	tf	200	0.7788	0.7587	0.7565
	tf-idf	200	0.7862	0.7642	0.7626

* Optimal hyperparameters in three classfiers. Refer to Table 1.

2.2 Task 4 and 5

Results of `CountVectorizer` with default values have already been shown in Table 3. Table 4 only presents the results after altering the parameters in `CountVectorizer`.

Table 4: Results of Task 4 and 5

Task	Classifier	Params.(t)*	Feature	Param.(clf)**	Precision	Recall	F1
4.a	NB	False	count	0.01	0.8136	0.8024	0.7881
			tf	0.0001	0.8255	0.8212	0.8213
			tf-idf	0.0001	0.8255	0.8212	0.8213
	SVM	False	count	0.1	0.7798	0.7726	0.7701
			tf	0.0001	0.8010	0.7958	0.7950
			tf-idf	0.0001	0.8044	0.7952	0.7941
	RF	False	count	200	0.7905	0.7681	0.7685
			tf	200	0.7769	0.7540	0.7535
			tf-idf	200	0.7748	0.7519	0.7509
4.b	**NB**	**"english"**	count	0.1	0.8139	0.8049	0.7900
			tf	**0.01**	**0.8405**	**0.8341**	**0.8348**
			tf-idf	**0.01**	**0.8405**	**0.8341**	**0.8348**
	SVM	"english"	count	0.1	0.8005	0.7921	0.7906
			tf	0.001	0.7950	0.7791	0.7760
			tf-idf	0.0001	0.8277	0.8214	0.8216
	RF	"english"	count	200	0.8039	0.7844	0.7836
			tf	200	0.8037	0.7831	0.7824
			tf-idf	200	0.8020	0.7857	0.7852
4.c	NB	"word", (1, 2)	count	0.01	0.8279	0.8123	0.8041

Task	Classifier	Params.(t)*	Feature	Param.(clf)**	Precision	Recall	F1
		"word", (1, 2)	tf	0.0001	0.8328	0.8280	0.8284
		"word", (1, 2)	tf-idf	0.0001	0.8328	0.8280	0.8284
	SVM	"word", (1, 2)	count	0.1	0.8075	0.8040	0.8028
		"word", (1, 2)	tf	0.0001	0.8224	0.8158	0.8152
		"word", (1, 2)	tf-idf	0.0001	0.8233	0.8176	0.8171
	RF	"word", (1, 2)	count	200	0.8022	0.7770	0.7772
		"word", (1, 2)	tf	150	0.7850	0.7554	0.7564
		"word", (1, 2)	tf-idf	150	0.7882	0.7608	0.7622
	NB	"char", (1, 1)	count	0.1	0.1705	0.1638	0.1514
		"char", (1, 1)	tf	0.0001	0.2165	0.1481	0.1166
		"char", (1, 1)	tf-idf	0.0001	0.2165	0.1481	0.1166
	SVM	"char", (1, 1)	count	0.001	0.2535	0.1793	0.1413
		"char", (1, 1)	tf	0.0001	0.3173	0.2383	0.2043
		"char", (1, 1)	tf-idf	0.001	0.2783	0.2177	0.1744
	RF	"char", (1, 1)	count	150	0.2897	0.2943	0.2813
		"char", (1, 1)	tf	100	0.2997	0.3114	0.2942
		"char", (1, 1)	tf-idf	200	0.3262	0.3304	0.3119
4.d	NB	115000	count	0.01	0.8085	0.7989	0.7846
		115000	tf	0.0001	0.8242	0.8199	0.8201
		115000	tf-idf	0.0001	0.8242	0.8199	0.8201
	SVM	115000	count	0.1	0.7802	0.7770	0.7747
		115000	tf	0.0001	0.8081	0.8002	0.7995
		115000	tf-idf	0.0001	0.8074	0.7983	0.7973
	RF	115000	count	200	0.7971	0.7728	0.7719
		115000	tf	200	0.7834	0.7643	0.7619
		115000	tf-idf	200	0.7795	0.7585	0.7570

* Hyperparameters in `CountVectorizer`. Refer to Table 2.
** Optimal hyperparameters in three classfiers. Refer to Table 1.

3 Discussion

According to results in Table 3 and Table 4, naïve bayes classifier with tf or tf-idf feature achieves the best performance on all metrics. Its performance can be improved further by setting stop_words parameter in `CountVectorizer` to "english". As stated in Table 4, the best precision, recall, and F1 are 0.8405, 0.8341, and 0.8348 respectively in our experiments.