Task 1

General speaking, we tried to use 3 different methods to transform the news text file into numeric vectors and using the figure data to train a model, and eventually predict binary categorical target.

Make text to be vectors :

1. After getting the dataset from 20newsgroup, which is a binary supervised classification to predict the categories of news, 'talk.politics.gun' or 'rec.sport.baseball'. 70% data for training and 30 % data for testing

2. We using three different methods, which are 'Bag-of-Word', 'TFIDF', 'word2vec' to do so
   - **Bag-of-words**
     - By counting how many times that the token shows up.
   - **TFIDF**
     - Term frequency-inverse document frequency(TFIDF) is a numerical statistic that is intended to reflect how important a word is to a document in collection or corpus.
   - **Word2vec**
     - Instead of getting the lexicon information of the text. Word2vec method extracting the representation or location in the corpus space by maximizing the concurrence probability.

Machine Learning Model with Feature Selections & Evaluation

We tried those combinations of data and models

1. Multinomial Naïve Bayes with Bag-of-Words features
2. Support Vector Machine with Bag-of-Words features
3. Support Vector Machine with TFIDF features
4. Support Vector Machine with Averaged word vector

- **Evaluation**
- We use self-generated function, 'get_metrics', to perform accuracy, precision, recall, F1 score of each model.

Model Evaluation Table

	MNB&BOW	SVM&BOW	SVM&TFIDF	SVW&AWV
Accuracy	96%	94%	94%	70%
Precision	95%	94%	95%	87%
Recall	96%	93%	93%	45%
F1 Score	96%	93%	94%	60%

- We use self-generated function, 'train_predict_evaluate_model', to perform predictions and evaluates the predications.

Confusion Matrix for 4 Models

	0	1
0	281	12
1	14	265

	0	1
0	278	15
1	29	250

	0	1
0	280	13
1	28	251

	0	1
0	240	53
1	36	243

MNB&BOW	SVM&BOW	SVM&TFIDF	SVW&AWV

But we want to improve the score of our performance.

Task 2

In this part I have tried Neural Network in order to outperform the former ones.

1. Multi-layer Neural Network

In task 2, we use packages sklearn.neural_network.MLPClassifier to implement Multi-layer Neural Network.

- **Multi-layer Neural Network with bag-of-words**
- **Multi-layer Neural Network with TFIDF**
- **Multi-layer Neural Network with Averaged Word Vector**

- **Evaluation- before tuning**

Model Evluation Table

	NN&BOW	NN&TFIDF	NN&AWV
Accuracy	92%	96%	83%
Precision	95%	98%	84%
Recall	86%	93%	77%
F1 Score	90%	95%	80%

- **Confusion Matrix**

Confusion Matrix for 3 Models

NN&BOW	NN&TFIDF	NN&AWV

	Actual_Sport	Actual_Politic	Actual_Sport	Actual_Politic	Actual_Sport	Actual_Politic
Predict_Sport	305	12	310	7	280	37
Predict_Politic	35	220	19	236	59	196

- **Result shows good in Neural Network.**
- **Tuning**
- **Evaluation- after tuning**
-

Model Evluation Table

	NN&BOW	NN&TFIDF	NN&AWV
Accuracy	92%	95%	82%
Precision	98%	97%	78%
Recall	83%	92%	83%
F1 Score	90%	95%	80%

- **Confusion Matrix**

Confusion Matrix for 3 Models

NN&BOW	NN&TFIDF	NN&AWV

	Actual_Sport	Actual_Politic	Actual_Sport	Actual_Politic	Actual_Sport	Actual_Politic
Predict_Sport	312	5	310	7	258	59
Predict_Politic	43	212	20	235	44	211

- **Result shows good in Neural Network.**

Task 3

In this part I have tried multiple classifiers in order to build a outperformed classifier than the former ones.

By ensemble a collection of classifiers including :

1. **Gaussian NB .**
2. **Perceptron**
3. **Logistics regression**
4. **MultiNB**
5. **Linear_SVM**
6. **SVM**
7. **Neural network**

After tuning all above and using majority vote to predict the final answer for BOW data :

Model Evluation Table

	Boo&BOW	Boosting&TFIDF
Accuracy	89%	84%
Precision	99%	100%
Recall	76%	64%
F1 Score	86%	78%

- **Confusion Matrix**

Confusion Matrix for 2 Models

Boosting&BOW		Boosting&TFIDF	
	Actual_Sport	**Actual_Politic**	**Actual_Sport**	**Actual_Politic**
Predict_Sport	316	1	317	0
Predict_Politic	61	194	93	162

This model perfect good at Precision ! It have unbeatable score on the field.

Limitation and Further Improvement:

1. **Text data might not be linear dividable ! According most non-linear classifier could reach better score. Try more non-linear classifiers in ensemble**
2. **Trying to treat different news instances with different weight, especially those instance usually putted in wrong set. (Adaboosting)**
3. **In the ensemble part give different weights for each classifiers instead of treating them with same voting right.**

These methods are focus on model side. We could also improve score by data preprocessing like :

    1. **W2V with larger dimension(features)**
    2. **Training vectors with bigger, better corpus.(more news)**

Detail could refer to the Jupiter notebook file.