What is the task?

Task: Classify new articles into various categories based on their content.

What ML solution did you choose and, most importantly, why was this an appropriate choice?

I tried two classifiers- Support Vector Classifier, and Naive Bayesian Classifier. The Support Vector Classifier worked the best.

Steps - Load data > Preprocess the data > Split train/dev/test > vectorize the feature and the target variables using TF-IDF vectorizer converting category to numbers > use test data in the inference process.

How did you choose to evaluate success?

I used the F1 score, accuracy score, Confusion matrix, and classification report to evaluate the models.

What software did you use and why did you choose it?

Language: Python as all the libraries.

Software: *scikit-learn* has all the machine learning algorithms prewritten, *NumPy, pandas, matplotlib, NLTK* and others that are written in python.

what are the results?

Predicting test data using Multinomial Naive Bayesian

	precision	recall	f1-score	support
0	0.938596	0.922414	0.930435	116.000000
1	0.832370	0.808989	0.820513	178.000000
2	0.834783	0.860987	0.847682	223.000000
accuracy	0.856867	0.856867	0.856867	0.856867
macro avg	0.868583	0.864130	0.866210	517.000000
weighted avg	0.857245	0.856867	0.856895	517.000000

```
0.8513513513513513
```

Predicting test data using Support Vector Machines

	precision	recall	f1-score	support
0	0.886792	0.903846	0.895238	52.000000
1	0.934211	0.788889	0.855422	90.000000
2	0.811966	0.913462	0.859729	104.000000
accuracy	0.865854	0.865854	0.865854	0.865854
macro avg	0.877656	0.868732	0.870129	246.000000
weighted avg	0.872507	0.865854	0.865659	246.000000

```
0.8744939271255061
```

The Support vector machine achieves a better result on the data. Hence it could be used in the production server.