Classification Model to distinguish posts from r/science and r/nature

By: Lin Junyuan

TABLE OF CONTENTS

01

Problem Statement

02

Data Gathering and EDA

03

Preprocessing & Modeling

04

Evaluation

05

Conclusion

Problem Statement

What is the

best classification model

that is able to identify whether a post belongs to the r/science subreddit or the r/nature subreddit with

at least 80% accuracy

and what are the

top 5 features?

Data Gathering and EDA

01 r/science

- 625 posts
- Set as 1 during encoding

02 r/nature

- 550 posts
- Set as 0 during encoding

- Extracted and merged [‘title’] and [‘selftext’] into a single column [‘post’]

Pre-processing and Modelling

01

Preprocessor

Lemmatizer
PorterStemmer

02

Transformer

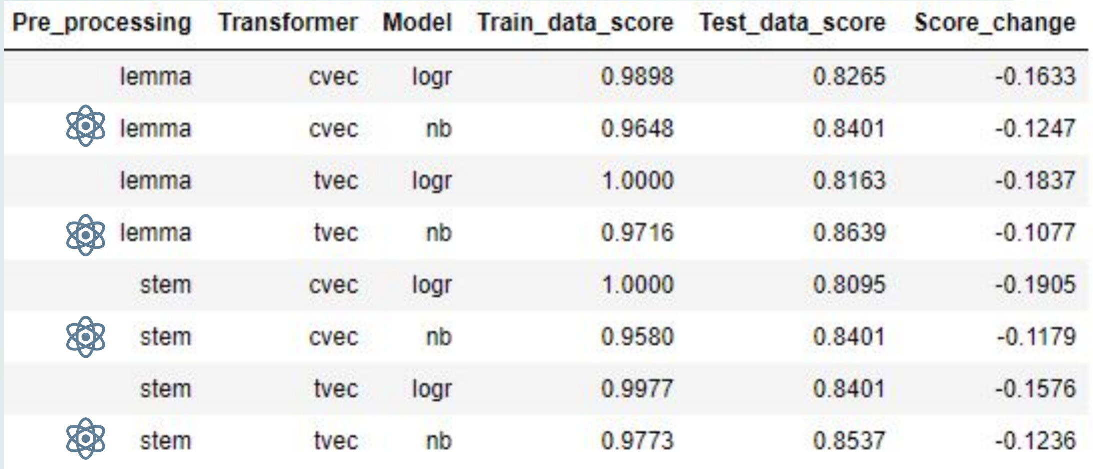
Count Vectorizor
TF-IDF Vectorizor

03

Model

Logistic Regression
Naive Bayes (MultinomialNB)

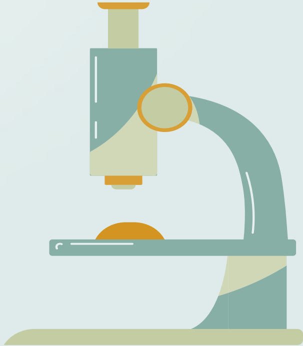
Model Evaluation



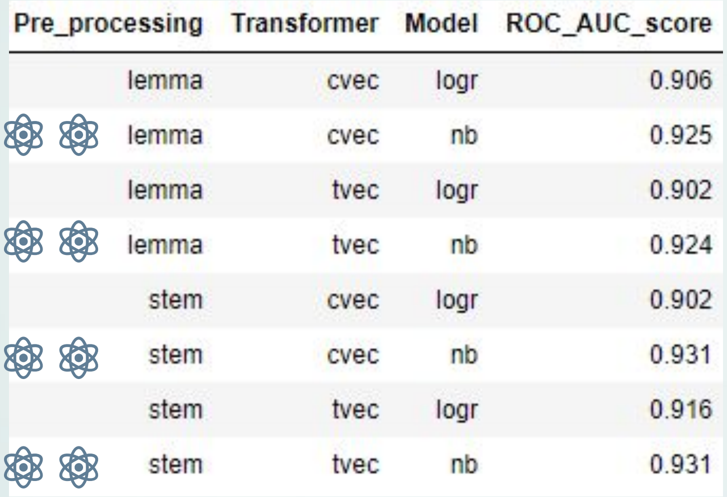
Pre_processing	Transformer	Model	Train_data_score	Test_data_score	Score_change
lemma	cvec	logr	0.9898	0.8265	-0.1633
⚛ lemma	cvec	nb	0.9648	0.8401	-0.1247
lemma	tvec	logr	1.0000	0.8163	-0.1837
⚛ lemma	tvec	nb	0.9716	0.8639	-0.1077
stem	cvec	logr	1.0000	0.8095	-0.1905
⚛ stem	cvec	nb	0.9580	0.8401	-0.1179
stem	tvec	logr	0.9977	0.8401	-0.1576
⚛ stem	tvec	nb	0.9773	0.8537	-0.1236

- All models show signs of overfitting due to high score on training data.

- Models with lower drop in score when tested with test data are marked with ⚛



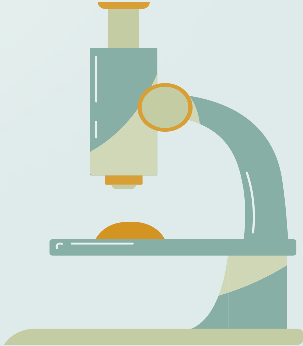
Model Evaluation



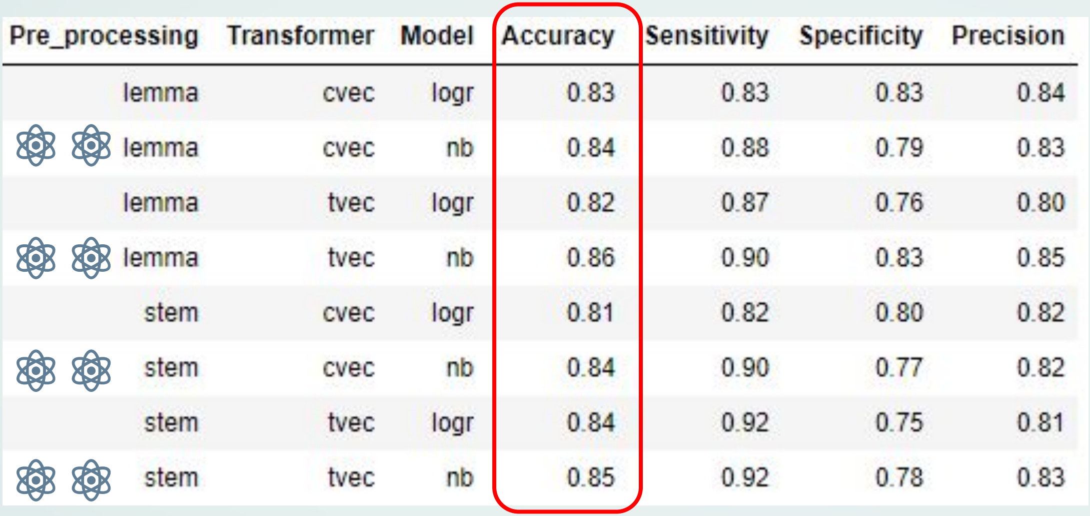
	Pre_processing	Transformer	Model	ROC_AUC_score
	lemma	cvec	logr	0.906
⚛ ⚛	lemma	cvec	nb	0.925
	lemma	tvec	logr	0.902
⚛ ⚛	lemma	tvec	nb	0.924
	stem	cvec	logr	0.902
⚛ ⚛	stem	cvec	nb	0.931
	stem	tvec	logr	0.916
⚛ ⚛	stem	tvec	nb	0.931

- All models have high ROC_AUC_scores

- Models with higher ROC_AUC_scores are marked with ⚛

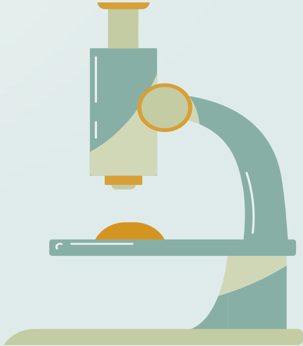


Model Evaluation

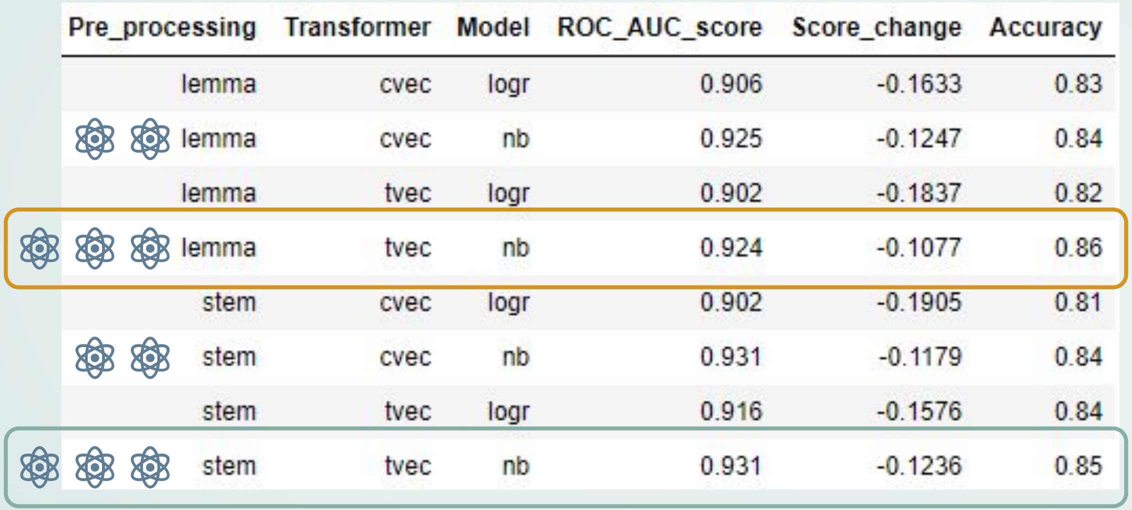


	Pre_processing	Transformer	Model	Accuracy	Sensitivity	Specificity	Precision
	lemma	cvec	logr	0.83	0.83	0.83	0.84
⚛ ⚛	lemma	cvec	nb	0.84	0.88	0.79	0.83
	lemma	tvec	logr	0.82	0.87	0.76	0.80
⚛ ⚛ ⚛	lemma	tvec	nb	0.86	0.90	0.83	0.85
	stem	cvec	logr	0.81	0.82	0.80	0.82
⚛ ⚛	stem	cvec	nb	0.84	0.90	0.77	0.82
	stem	tvec	logr	0.84	0.92	0.75	0.81
⚛ ⚛ ⚛	stem	tvec	nb	0.85	0.92	0.78	0.83

- Accuracy is chosen as the main metric.
- Models with higher Accuracy are marked with ⚛



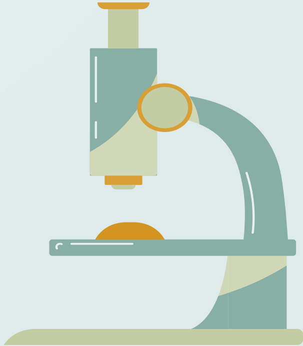
Model Evaluation



	Pre_processing	Transformer	Model	ROC_AUC_score	Score_change	Accuracy
	lemma	cvec	logr	0.906	-0.1633	0.83
⚛⚛	lemma	cvec	nb	0.925	-0.1247	0.84
	lemma	tvec	logr	0.902	-0.1837	0.82
⚛⚛⚛	lemma	tvec	nb	0.924	-0.1077	0.86
	stem	cvec	logr	0.902	-0.1905	0.81
⚛⚛	stem	cvec	nb	0.931	-0.1179	0.84
	stem	tvec	logr	0.916	-0.1576	0.84
⚛⚛⚛	stem	tvec	nb	0.931	-0.1236	0.85

lemma/tvec/nb model chosen due to:

- Slightly higher accuracy
- Lower drop in score change
- Lemmatized words are more meaningful



Top 5 features of selected model

01 Study

02 Covid

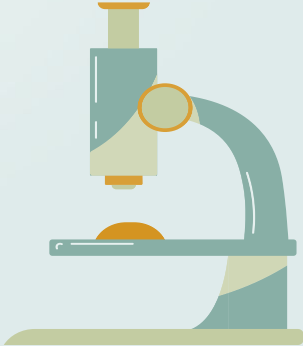
03 New

04 Researcher

05 People

Model Evaluation

- Tested model on unseen data
- Extracted another 100 posts from each subreddit and ran predictions
- Model was able to score 86% accuracy



Conclusion

- Lemmatization -
  TF-IDF Vectorizer -
  Naive Bayes MultinomialNB

  model found to be best suited to the problem statement.

- Top 5 features are
  - Study
  - Covid
  - New
  - Researcher
  - People

Recommendations

- Due to ‘Covid’ being one of the top features, it indicates that model is picking up certain time-sensitive terms.
    - Can be mitigated by building up training data over a period of time and reinforce the model

- High tendency of overfitting to training data
    - Modify GridSearch to look for parameters that lead to least drop in score when testing model on test data