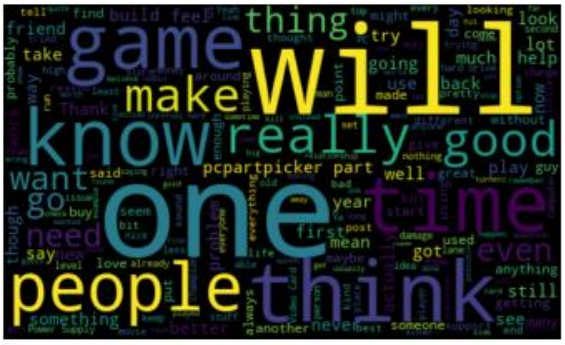
Text-as-Data Coursework

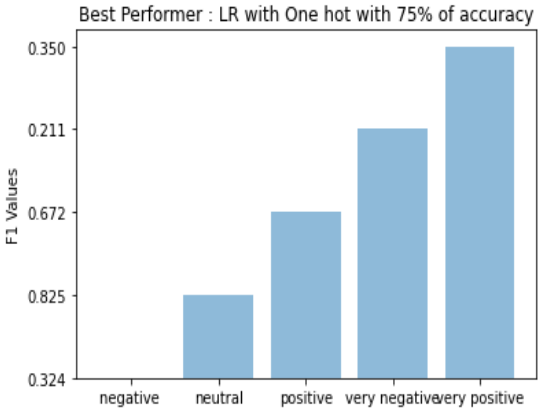
Collab notebook link : https://colab.research.google.com/drive/1A4MzyevjKG6Y63ckefhqokULR1VQ6LRh?usp=sharing

Q1

i) Results

	CLASSIFIER		VALIDATION DATA			
			ACC	PRECISION	RECALL	F1
1	Dummy Classifier with strategy="most_frequent		0.631	1	0.631	0.774
		MACRO AVG		0.2	0.12	0.15
2	Dummy Classifier with strategy="stratified"		0.478	0.478	0.478	0.478
		MACRO AVG		0.213	0.210	0.211
3	LogisticRegression with One-hot vectorization		0.748	0.79	0.752	0.766
		MACRO AVG		0.461	0.633	0.515
4	LogisticRegression with TF-IDF vectorization		0.609	0.896	0.61	0.772
		MACRO AVG		0.2	0.182	0.164
5	SVC Classifier with One-hot vectorization		0.722	0.859	0.722	0.773
		MACRO AVG		0.285	0.423	0.284
6	An 'interesting' classifier (CATBOOST CLASSIFER)		0.727	0.838	0.727	0.764
		MACRO AVG		0.338	0.727	0.764
	CLASSIFIER		**TEST DATA**			
			ACC	PRECISION	RECALL	F1
1	Dummy Classifier with strategy="most_frequent		0.625	1	0.626	0.77
		MACRO AVG		0.2	0.125	0.154
2	Dummy Classifier with strategy="stratified"		0.475	0.481	0.475	0.478
		MACRO AVG		0.196	0.197	0.197
3	LogisticRegression with One-hot vectorization		0.741	0.787	0.748	0.763
		MACRO AVG		0.432	0.622	0.476
4	LogisticRegression with TF-IDF vectorization		0.609	0.923	0.609	0.729
		MACRO AVG		0.199	0.181	0.165
5	SVC Classifier with One-hot vectorization		0.73	0.875	0.73	0.782
		MACRO AVG		0.288	0.458	0.287
6	An 'interesting' classifier (CATBOOST CLASSIER)		0.728	0.834	0.728	0.765
		MACRO AVG		0.47	0.487	0.478





- Assumptions made from distribution of labels are “body” is review text which we have to analyse and work with, we have “sentiment. Polarity” which says whether review given by the user is negative – positive. “Majority_type” can be used analyse the review under which category it comes like it is a question or appreciation etc.
  Each train, validation and test data set have 12138 rows × 12 columns, 3109 rows × 12 columns, 4016 rows × 12 columns respectively
  Top sentiment.polarity:
  neutral 7679
  positive 3231
  negative 878
  very positive 253
  very negative 97

- Pre-processing techniques are 1) Making words to lower 2) Removing punctuations numbers and special characters. 3)Removing short words 4) Tokenize and Normalize

ii) Analysis and discussion

- All classifiers produced accuracy of over 60% minimum in both validation and test data sets while LR – one hot and SVC-one hot were able to produce accuracy of 75% and 73% respectively. While Dummy Classifier with strategy="stratified" gave only 47%. Processing techniques involved include tokenization, normalization, and vectorization.

- Classifier I used is Decision tree classifier with maximum entropy technique, I used spacy for tokenization and normalization and the vectorization technique used is one hot vectorization since it gives better accuracy than TF-IDF. While using this classifier I was able to get 70% accuracy for both test and validation sets. random_state ,max_depth are the parameters used .Even though it was unable to produce better accuracy than LR-one hot it was able to produce 70% .

	CLASSIFIER	Key Parameters
1	Dummy Classifier with strategy="most_frequent	strategy
2	Dummy Classifier with strategy="stratified"	strategy
3	LogisticRegression with One-hot vectorization	default
4	LogisticRegression with TF-IDF vectorization	C
5	SVC Classifier with One-hot vectorization	Kernel
6	An ‘interesting’ classifier (DECISION TREE CLASSIER)	Random state and max depth

- Logistic regression and SVC with one hot was able to perform better than TF-IDF. From the dataset we basically considered only 2 labels i.e. body and sentiment.polarity.

Q2

	CLASSIFIER		VALIDATION DATA			
			ACC	PRECISION	RECALL	F1
1	Logistic Regression with TF-IDF post tuning the params		0.631	1	0.631	0.774
		MACRO AVG		0.2	0.126	0.155
	CLASSIFIER		**TEST DATA**			
			ACC	PRECISION	RECALL	F1
1	Logistic Regression with TF-IDF post tuning the params		0.626	1	0.626	0.77
		MACRO AVG		0.2	0.125	0.154

Doing these parameter tuning I was able to increase the accuracy by **60% to 63%**
Parameters tried in classifier:

(a) **Penalty:** Found no change in accuracywhen changing it to 'elasticnet' as I used 'saga' as optimizer
(b) **C:** Found big difference while changing the C value since smaller the values strong the regularization and more accuracy
(c) **Solver:** Found NO noticeable difference between 'sag' , 'saga' and 'newton-cg'
(d) **max_iter:** Very less variations in the accuracy with the number

Parameters tried in vectorizer:

(a) **stop_words:** Found no change in accuracy when changed from default value
(b) **ngram_range:** Found considerable increase in the accuracy when changed it to both unigrams and bigrams
(c) **max_features:** Increased the accuracy when introduced as params
(d) **sublinear_tf:** Increased the accuracy when introduced as params and make it to TRUE

Error Analysis:
There are multiple types of errors associated with machine learning and predictive analytics. The primary types are in-sample and out-of-sample errors. In-sample errors (aka re-substitution errors) are the error rate found from the training data, i.e., the data used to build predictive models.

Out-of-sample errors (aka generalisation errors) are the error rates found on a new data set, and are the most important since they represent the potential performance of a given predictive model on new and unseen data.

In-sample error rates may be very low and seem to be indicative of a high-performing model, but one must be careful, as this may be due to overfitting as mentioned, which would result in a model that is unable to generalise well to new data.

Training and validation data is used to build, validate, and tune a model, but test data is used to evaluate model performance and generalisation capability. One very important point to note is that prediction performance and error analysis should only be done on test data, when evaluating a model for use on non-training or new data (out-of-sample).

Q3

	CLASSIFIER		TEST DATA			
			ACC	PRECISION	RECALL	F1
1	Logistic Regression with Count vector post tuning the params		0.747	0.799	0.747	0.77
		MACRO AVG		0.4	0.7	0.448

Confusion matrix:
[[62 214 6 0 0]
[55 2242 209 0 8]
[2 415 673 0 12]
[6 24 0 2 0]
[0 24 41 0 21]]
New features included is changing vectorization method from TF IDF to Count vectorization with one hot vectorization.
Changing this gave good accuracy change from 63% to 75%.
Hence I consider Countvector as better method for vectorization.
I implemented same to training, validation and testing data.

Another change which I considered was changing multinomial classes in polarity to binomial like

Neutral, Positive, Very Positive => 1
Negative, Very Negative => 0

This will increase the accuracy drastically to up to 93%

	CLASSIFIER		TEST DATA			
			ACC	PRECISION	RECALL	F1
1	Logistic Regression with Count vector post tuning the params		0.937	0.971	0.937	0.950
		MACRO AVG		0.624	0.819	0.671

We can also add pipeline method by adding “body” – review text and majority_type and then we can use this pipeline to predict the polarity value this will increase the accuracy.