

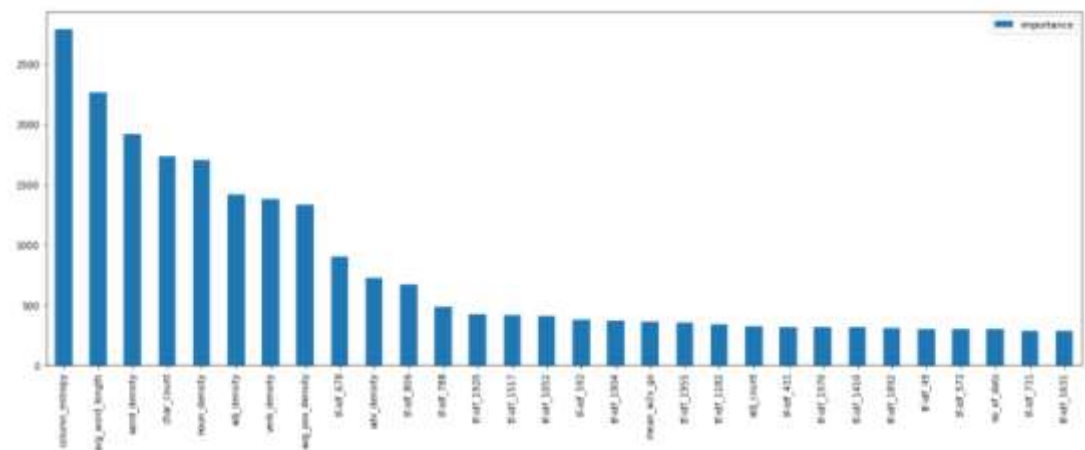
## 2. Approach-I:

## Traditional Machine Learning-Based

- i. I have used the following categories of feature engineering:
  - a. POS-based count and density
  - b. Static features capturing special character count
  - c. TF-IDF
  - d. Word2Vec
- ii. For Categorization, I have used LGBM as the final algorithm.

	precision	recall	f1-score
1	0.97	0.93	0.95
2	0.95	0.85	0.89
3	0.95	0.99	0.97
4	0.72	0.86	0.79
5	0.74	0.94	0.83
6	0.79	1.00	0.88
7	0.82	1.00	0.90
accuracy			0.90
macro avg	0.85	0.94	0.89
weighted avg	0.91	0.90	0.90

- iii. Train Performance:
- iv. Feature Importance



```
** We can observe in top-30 Features:
```

- v. Test Performance: 84%

### 3. Approach-II:

### Transformer-based Text Classification.

- i. It's a deep learning-based solution.
- ii. I dumped all the information in one column and used it as an input to the model.
- iii. Train Performance F1 'weighted' : 91%
- iv. Test Performance F1 'weighted' : 85%
- v. However, I decided to go with this as my final model because while training the difference between train and validation was the least for this model.