LING 406 Assignment #3 Report
Spring 2020 Hongshuo Zhang

I tried using LabelEncoder and POS_encoder to encode the dataset.
Clearly POS_encoder works better.
LabelEncoder → Each unique word gets a unique number.
POS_encoder → All words in a unique POS get a unique number.

Tables: (NB) Naive Bayes, (DT) Decision Tree
ATTR: Attribute in %, PRE: Precision, REC: Recall, ACC: accuracy

NB with LabelEncoder	F1 (%)	F1 ATTR	PRE (%)	PRE ATTR	REC (%)	REC ATTR	ACC (%)	ACC ATTR
All	19.7	N/A	21.0	N/A	22.3	N/A	33.0	N/A
Removed ‘a1’	20.6	-0.9	22.6	-1.6	23.1	-0.8	33.7	-0.7
Removed ‘a2’	19.5	+1.1	20.1	+2.5	22.2	+0.1	33.4	-0.4
Removed ‘a3’	19.9	-0.2	20.5	+0.5	22.8	-0.5	33.2	-0.2
Removed ‘a4’	5.0	+14.7	4.8	+16.2	7.4	+14.9	19.4	+13.6
Removed ‘a5’	19.6	+0.1	21.2	-0.2	22.6	-0.3	32.2	+0.8
Removed ‘a6’	20.3	-0.6	21.1	-0.1	23.0	-0.7	33.2	-0.2
Removed ‘a7’	19.7	0.0	21.2	-0.2	22.1	+0.2	32.8	+0.2

ATTR: Attribute in %, PRE: Precision, REC: Recall

DT with LabelEncoder	F1 (%)	F1 ATTR	PRE (%)	PRE ATTR	REC (%)	REC ATTR	ACC (%)	ACC ATTR
All	52.9	N/A	54.3	N/A	54.8	N/A	60.9	N/A

Removed ‘a1’	53.7	-0.8	54.9	-0.6	55.6	-0.8	61.2	-0.3
Removed ‘a2’	54.1	-1.2	54.8	-0.5	56.4	-1.6	61.1	-0.2
Removed ‘a3’	53.7	-0.8	54.3	0.0	55.8	-1.0	61.0	-0.1
Removed ‘a4’	12.7	+40.2	13.3	+41.0	13.1	+41.7	22.1	+38.8
Removed ‘a5’	56.9	-4.0	57.9	-3.6	58.8	-4.0	62.0	-1.9
Removed ‘a6’	54.9	-2.0	56.1	-1.8	56.7	-1.9	61.7	-0.8
Removed ‘a7’	54.8	-1.9	56.5	-2.2	55.8	+1.0	61.2	-0.3

Tables: (NB) Naive Bayes, (DT) Decision Tree
ATTR: Attribute in %, PRE: Precision, REC: Recall

NB with POS Encoder	F1 (%)	F1 ATTR	PRE (%)	PRE ATTR	REC (%)	REC ATTR	ACC (%)	ACC ATTR
All	65.8	N/A	68.0	N/A	67.4	N/A	79.2	N/A
Removed ‘a1’	67.0	-1.2	69.3	-1.3	68.5	-0.9	80.6	-1.4
Removed ‘a2’	65.3	+0.5	66.5	+1.5	67.2	+0.2	79.5	-0.3
Removed ‘a3’	65.0	+0.8	66.3	+1.7	66.4	+1.0	79.1	+0.1
Removed ‘a4’	6.1	+59.7	6.9	+61.1	9.7	+57.7	19.1	+60.1
Removed ‘a5’	64.7	+1.1	66.1	+1.9	66.5	+0.9	79.8	-0.6
Removed	65.8	+0.0	67.0	+1.0	67.7	-0.3	79.3	-0.1

‘a6’								
Removed ‘a7’	65.7	+0.1	67.0	+1.0	67.5	-0.1	79.2	0.0

Tables: (NB) Naive Bayes, (DT) Decision Tree
ATTR: Attribute in %, PRE: Precision, REC: Recall

DT with POS Encoder	F1 (%)	F1 ATTR	PRE (%)	PRE ATTR	REC (%)	REC ATTR	ACC (%)	ACC ATTR
All	85.5	N/A	86.0	N/A	87.2	N/A	94.9	N/A
Removed ‘a1’	83.5	+2.0	84.2	+1.8	85.2	+0.2	95.0	-0.1
Removed ‘a2’	86.5	-1.0	86.6	-0.6	88.0	-0.8	95.1	-0.2
Removed ‘a3’	84.7	+0.8	84.6	+1.4	86.3	+0.9	94.1	+0.8
Removed ‘a4’	18.0	+67.5	18.9	+67.1	18.6	+68.6	32.0	+62.9
Removed ‘a5’	84.8	+0.7	85.1	+0.9	86.0	+1.2	94.4	+0.5
Removed ‘a6’	86.9	-1.4	86.9	-0.9	89.1	-1.9	95.5	-0.6
Removed ‘a7’	85.3	+0.2	85.0	+1.0	87.3	-0.1	95.1	-0.2

Q: What is the best machine learning algorithm (classifier) for this task? You need to discuss this per metric used to compute the performance.

The best machine learning algorithm (classifier) is undoubtedly the decision tree model. After trying encoding the dataset with the LabelEncoder, it achieved around 20% of precision, recall and F-1 with the naive bayes model. However, the decision tree model achieves a precision, recall and F-1 of roughly 60%. This is a substantial improvement from these metrics from the bayes model. Therefore, based on the empirical evidence, the decision tree model is better than the naive bayes model.

The same held true while I was running the POS-encoded dataset against these two models. It achieved around 65% of precision, recall and F-1 with the naive bayes model and 85% of precision, recall and F-1with the decision tree model. The decision tree model is ahead of the naive bayes model, in performance, by roughly 20%!

Q: Which features contributed the most to the performance? Which contributed to the least?

Naive Bayes:

Feature ‘a4’ contributed the most to the performance.

Feature ‘a1’ contributed the least to the performance. (It often contributed negatively, according to the table in the first two pages.)

Decision Tree:

Feature ‘a4’ contributed the most to the performance.

Feature ‘a5’ contributed the least to the performance. (It often contributed negatively, according to the table in the first two pages.)

Q: How good is the feature set?

The feature set is not good at all. Since we already know that feature ‘a4’ already contributed so much in each round. (up to 70%!!!!). It is entirely possible that we only train the model with the ‘a4’ column alone. In fact, I have already tried doing it and it gives me better results. This means that pretty much all other columns are kinda useless. If the most of the weight is on feature ‘a4’, then the feature set would not be very useful since we not only want to know what part of speech tag should be assigned for a given word alone, we also want to know what part of speech tag should be assigned for a given word in a given context!

Q: What attributes would I choose?

As I mentioned in the previous problem, it is entirely possible to train the model with ‘a4’ alone. Sometimes if I choose to train the model with feature ‘a4’ alone, the models could potentially give me better performance. Isn’t that ironic that most other features don’t contribute so much, sometimes often contribute negatively? Not only that, training with more features also puts a heavier load to the system since it takes more time to train and it eats up more system memory. So, I would leave only 2 or 3 features in a dataset, and remove all others.