CSE 537 Artificial Intelligence
Project 5

Hsiang Yu Cheng	**111322987**	**Kishan Nerella**	**111493601**
Naresh Nalam	**111482942**	**Rakshith Reddy**	**111498989**

Question 1

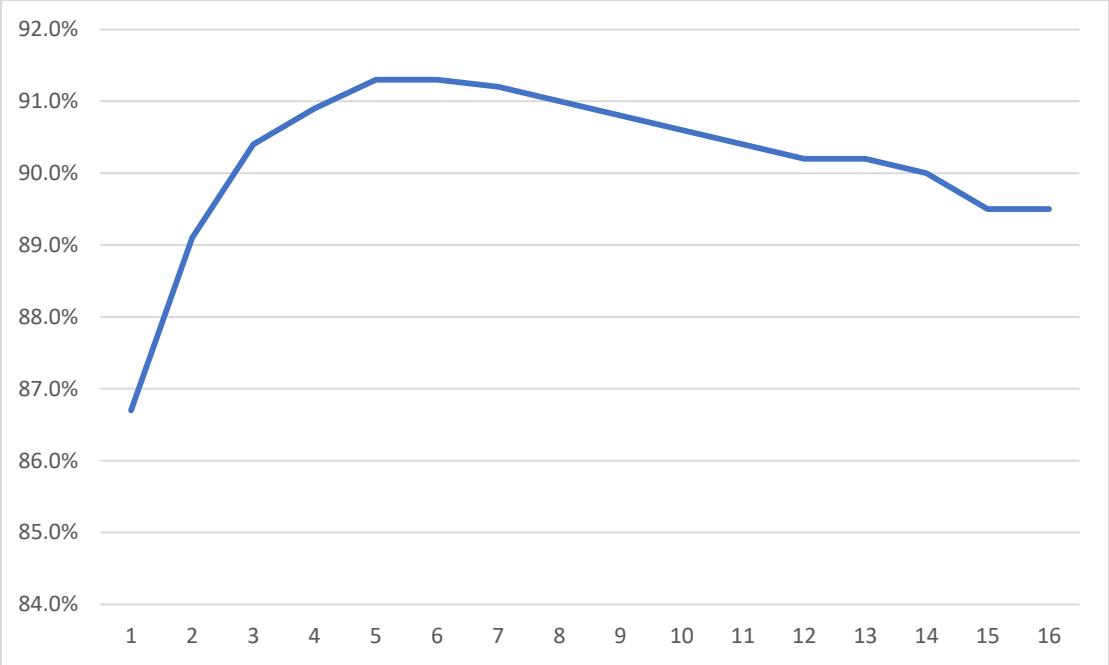
P_val	Accuracy	Nodes
0.01	73.184%	391
0.05	73.22%	991
1.00	69.716%	26711

As p_val nears 1, we can see that the number of nodes is quite high because we're never stopping to split. Also, the accuracy is low for p_val = 1 since it overfits the training the data and doesn't perform well on test data. There is no much difference in accuracy when p_val is 0.01 or 0.05. This means that the upper level nodes are deciding the label and lower level nodes are not that significant in determining the label. If we want to choose one p_val, we would choose p_val 0.01 since it giving the same accuracy as 0.01 but has less nodes which makes the prediction faster.

Question 2

The best parameter is between 41 and 51

Smoothing parameter	accuracy
1	86.7%
11	89.1%
21	90.4%
31	90.9%
41	91.3%
51	91.3%
61	91.2%
71	91.0%
81	90.8%
91	90.6%
101	90.4%
111	90.2%
121	90.2%
131	90.0%
141	89.5%



Extra Credit

We implemented a class “SpecialRules” which remaps words to certain string if they are match by some rules. This class maintains a List() object storing rules in compiled re() objects, and match input string against these rule accordingly to the order they are added to the class. If anything is matched, it looks up a Dict() object for corresponding string to map to. It does not make much difference in this point because we are not able to implement efficient rules in time.

We have thought of utilizing rules such as remapping every occurrence of verbs in different conjugation. However, this requires extracting data from dictionaries from online resources, and for the reason that we have to prepare for our final exams and contacting professors for projects involving CSE523/524, we are not able to accomplish this task before deadline.

With the insight that this may be a suitable approach, we decided to keep the implementation in the code we submitted for future uses. Currently it has only limited functionalities, however, it is possible to be inherited by more complex rule handlers, for instance, something that handles nested rules or maps input string to multiple different strings. Looking forward to sometime in the future, there is someone can put this concept in use for developing a better spam filter.

Contribution

Hsiang and Rakshith worked on Question 2.

Kishan and Naresh worked on Question 1.