



# Bad PROJECT 3

**PREPARED FOR**

Dr. Sabine Bergler

**PREPARED BY**

Karandeep

40104845

## TABLE OF CONTENTS

<b>Introduction:</b>	<b>2</b>
Purpose	2
<b>Good Validation Data (Feature Context Free Grammar)/ Sentiment Parser</b>	<b>3</b>
Examples	3

## Introduction:

**About NLTK** - NLTK aka the Natural Language Toolkit, is a suite of open source Python modules, data sets, and tutorials supporting research and development in Natural Language Processing. It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning. It also includes graphical demonstrations and sample data sets as well as accompanied by a cook book and a book which explains the principles behind the underlying language processing tasks that NLTK supports.

## Purpose:

Hidden information often lies deep within the boundaries of what we can perceive with our eyes and our ears. Some look to data for that purpose, and most of the time, data can tell us more than we thought was imaginable. But sometimes data might not be clear cut enough to perform any sort of analytics. Language, tone, and sentence structure can explain a lot about how people are feeling, and can even be used to predict how people might feel about similar topics using a combination of the Natural Language Toolkit, a Python library used for analyzing text, and machine learning.

For this project I was to develop a Feature grammar to parse given sentences and judge the sentiment behind the sentence. The console has options to run **Sentiment Analyzer** either on a good inbuilt data or bad inbuilt data sentences to parse. Analysis of data has been done in a way that is easily comprehensible. Overall goal of the project was to create a parser and a FCFG which filters data by using NLTK access commands and creates sentence trees and visualize them in a more informative way. NLTK is not perfect and lacks in some areas which is being explored below.

# Bad Validation Data (Feature Context Free Grammar)/ Sentiment Parser:

## Examples:

- Good Sentence 1

OUTPUT:

```
Sentence -> this movie is amazingly awful
(S[-INV]
 (NP[] (NP[] (DT[] this)) (NP[+WH] movie))
 (VP[]
  (V[+AUX] is)
  (VP[]
   (NP[]
    (NP[] (ADJ[SNT='POS'] amazingly))
    (NP[] (ADJ[SNT='NEG'] awful))))))

#####

Sentence is neutral

#####
```

- GoodSentence 2

OUTPUT:

```
Sentence -> it is pretty disgusting
(S[-INV]
 (NP[+WH] it)
 (VP[]
  (V[+AUX] is)
  (VP[]
   (NP[]
    (NP[] (ADJ[SNT='POS'] pretty))
    (NP[] (ADJ[SNT='NEG'] disgusting))))))

#####

Sentence is neutral

#####
```

- Good Sentence 3

OUTPUT:

```
Sentence -> it is not a good movie
(S[-INV]
 (NP[+WH] it)
 (VP[]
  (V[+AUX] (V[+AUX] is) (Adv[+NEG] not))
  (NP[]
   (NP[] (NP[] (DT[] a)) (NP[] (ADJ[SNT='POS'] good)))
   (NP[+WH] movie))))
```

#####

Sentence is neutral

#####

- Good Sentence 4

OUTPUT:

```
Sentence -> this does not have a compelling factor in it
(S[-INV]
 (NP[] (DT[] this))
 (VP[]
  (V[+AUX] (V[+AUX] does) (Adv[+NEG] not))
  (VP[]
   (V[+AUX] have)
   (VP[]
    (NP[]
     (NP[]
      (NP[] (NP[] (DT[] a)) (NP[] (ADJ[SNT='POS'] compelling)))
      (NP[+WH] factor))
     (NP[] (PRP[] in) (NP[+WH] it))))))
```

#####

Sentence is neutral

#####

- Good Sentence 5

OUTPUT:

```
Sentence -> it is not a bad movie
(S[-INV]
 (NP[+WH] it)
 (VP[]
  (V[+AUX] (V[+AUX] is) (Adv[+NEG] not))
  (NP[]
   (NP[] (NP[] (DT[] a)) (NP[] (ADJ[SNT='NEG'] bad)))
   (NP[+WH] movie))))
```

#####

Sentence is negative.

#####