NAME : Dhyan

REGNO: 20BCE7648

NLP - DIGITAL ASSIGNMENT

Q1] Given context - free grammar is:

S → NP VP

NP → DP

DP → DET NN

DET → a / the

NN → car / bike

VP → is VRB

VRB → running / stopping

Shift Reduce Parser.

Stack	Input	Action
$	a car is running $	shift
$ a	car is running $	Reduce (Det → a)
$ Det	car is running $	shift
$ Det car	is running $	Reduce (NN - car)
$ Det NN	is running $	Reduce (DP - DETNN)
$ DP	is running $	shift
$ DP is	running $	shift
$ DP is running.	$	Reduce (VP → DP)
$ NP is running	$	Reduce (VRB → running)
$ NP is VRB	$	Reduce (VP → is VRB)
$ NP VP	$	Reduce (S → NP VP)
$ S		ACCEPT

END

classmate

* The Parser successfully accepts the input string "a car is running" which means that it can be generated by the given grammar.

PYTHON CODE

```
import nltk

grammar = nltk.CFG.fromString("""
    S → NP VP
    NP → PP
    PP → DET NN
    Det → 'a' | 'the'
    NN → 'car' | 'bike'
    VP → 'is' VRB
    VRB → 'running' | 'stopping'
""")

# Generate all possible sentences

sentences = []
for length in range(1, 6):
    for tree in grammar.generate(n = 1000, depth = length):
        sentence = ' '.join(tree.leaves())
        sentence.append(sentence)
print(sentences)
```

Q2]

$E(\text{Play Golf}) = E(7,4)$

$= -7/11 \log 7/11 - 4/11 \log 4/11$

$= 0.45 + 0531 = \underline{0.946}$

$E(\text{Play Golf, Temp}) = 3/11\, E(1,2) + 4/11\, (3,1) + 4/11\, E(3,1)$

$= 3/11\, 1 - 1/3 \log 1/3 - 2/3 \log 1/3) +$
$3/11\, (-3/4 \log 3/4 - 1/4 \log 1/4)$

$= 0.2504 + 0.59 = \underline{0.8464}$

$E(\text{play golf, humidity}) = 5/14\ E(2,3) + 5/11\ (5,11)$

$= 5/11\left(-2/5 \log 2/5 - 2/5 \log 3/5\right)$

$+ 6/11\ \left(-5/6 \log 5/6 - 1/6 \log 1/6\right)$

$= 0.44 + 0.35 = 0.7458$

$E(\text{Play Golf, Windy}) = 4/11\ (E\ 2,2) + 7/11 E(5,2)$

$= 4/11\ (1) + 7/11\ \left(-5/9 \log 5/9 + 2/4 \log 2/4\right)$

$= 0.9128$

Gain (Temp) = 0.1056
Gain (Humidity) = 0.1502
Gain (Windy) = 0.332

Humidity

High — Normal.

High Humidity

$$E(\text{High}) = -\tfrac{2}{5}\log\tfrac{2}{5} - \tfrac{3}{5}\log\tfrac{3}{5} = 0.971$$

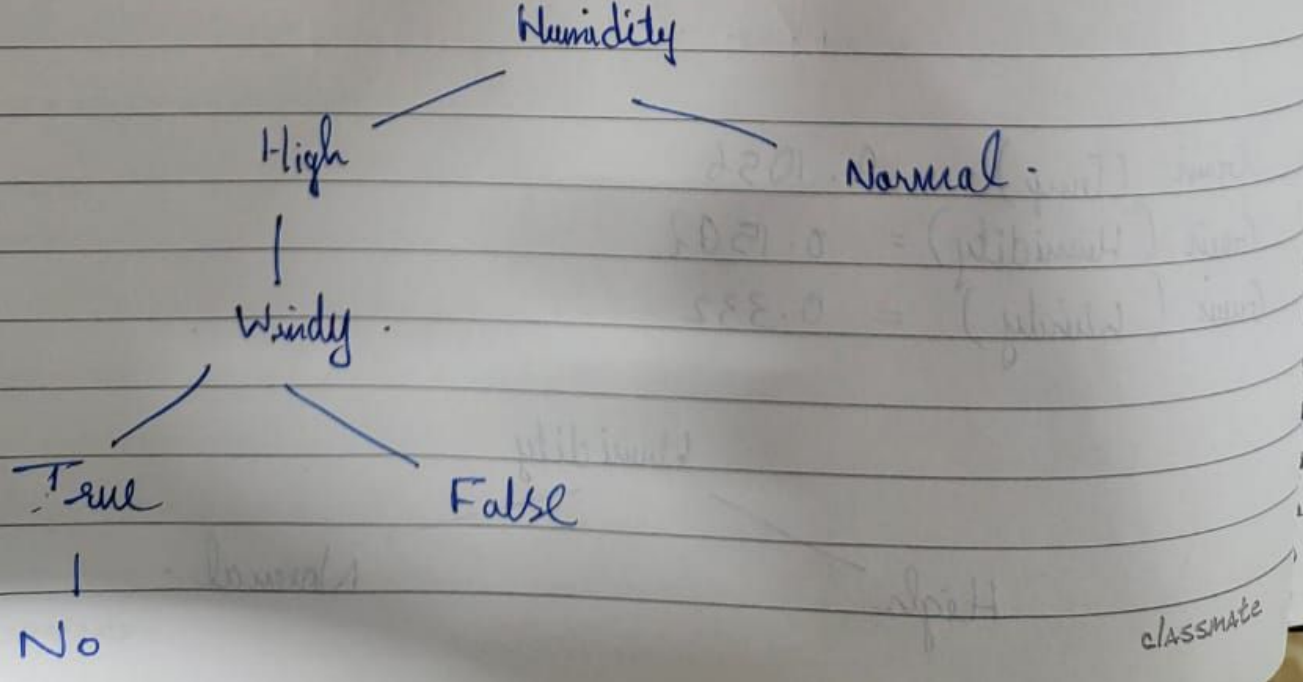
$$E(\text{Golf}, \text{Temp}) = \tfrac{3}{5}\left(-\tfrac{1}{3}\log\tfrac{1}{3} - \tfrac{2}{3}\log\tfrac{2}{3}\right) + \tfrac{2}{5}(1) + 0$$

$$= 0.551 + 0.4 = 0.951$$

$$E(\text{Golf}, \text{Windy}) = \tfrac{1}{5}(0) + \tfrac{4}{5}(1) = 0.8$$

Gain (Temp) = 0.02

Gain (Windy) = 0.171



Normal

E (Normal) = 0.6499.

E(Golf, Temp) = 2/6 0 + 4/6 (-3/4 log 3/4 - 1/4 log 1/4)

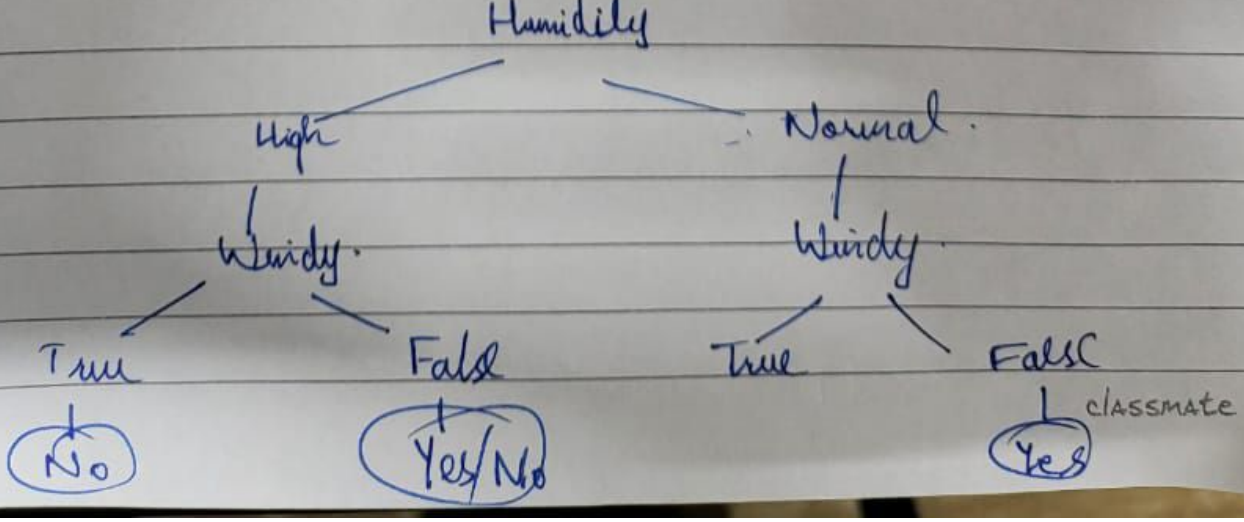
= 0.5408

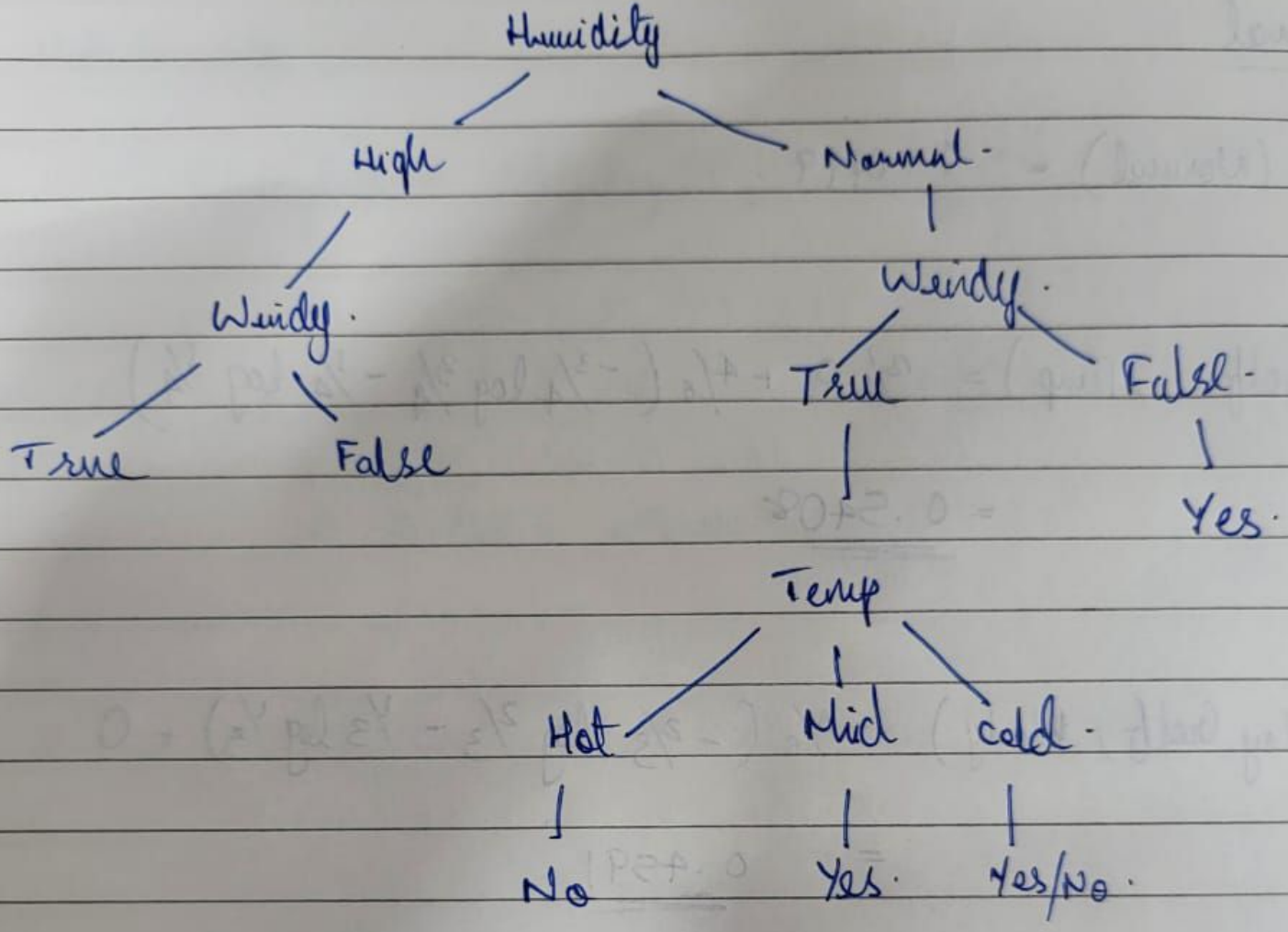
E (Play Golf, Windy) = 3/6 (-2/3 log 2/3 - 1/3 log 1/3) + 0

= 0.4591

Gain (Temp) = 0.1091

Gain (Windy) = 0.1908





P(Play Golf) = Yes.

Q3] CFG :

S → NP VP
NP → Det NN
Det → a / an / the
NN → child / adult
VP → AUX VRB
AUX → is / was
VRB → crying / sleeping.

Stack	Input	Action
$	a child is crying $	shift
$ a	child is crying $	Reduce (Det → a)
$ Det	child is crying $	shift
$ Det child	is crying $	Reduce (NN → child)
$ Det NN	is crying $	Reduce (NP → Det NN)
$ NP	is crying $	shift
$ NP is	crying $	Reduce (AUX → is)
$ NP AUX	crying $	shift
$ NP AUX crying	$	Reduce (VRB → crying)
$ NP AUX VRB	$	Reduce (VP → AUX VRB)
$ NP VP	$	Reduce (S → NP VP)
$ S		ACCEPT

⟹ Parser successfully accepts "a child is crying"

Code - PYTHON

```
import nltk

grammar = nltk.CFG.fromString("""
    S --> NP VP
    NP --> Det NN
    Det --> 'a' | 'an' | 'the'
    NN --> 'child' | 'adult'
    VP --> AUX VRB
    AUX --> 'is' | 'was'
    VRB --> 'crying' | 'sleeping'

sr_parser = nltk.ShiftReduceParser(grammar)

sentence = "a child is crying"
tokens = nltk.word_tokenize(sentence)

try:
    for tree in sr_parse.parse(tokens):
        print(tree)
    print("Sentence is grammatically correct")

except ValueError:
    print("Sentence is grammatically incorrect")
```

classmate

Q4] $P(No) = 4/10$

$P(Yes) = 6/10$

Now, calculate probability of likelyhood of evidence · Given ~~child, young, low~~.

$X = \{$ type of family = single parent
age group = young
income_status = high. $\}$.

$P(\text{Single Parent}/Yes) = 1/6$

$P(\text{Single Parent}/No) = 1/4$

$P(\text{Young}/Yes) = 2/6$

$P(\text{Young}/No) = 1/4$

$P(\text{Low}/Yes) = 1/6$; $P(\text{High}/Yes) = 2/2$

$P(\text{Low}/No) = 4/4$; $P(\text{High}/No) = 0$

$$P(\text{Yes}/X) = P\left(\frac{\text{Single Parent}}{\text{Yes}}\right) * P\left(\frac{\text{Young}}{\text{Yes}}\right) * P\left(\frac{\text{High}}{\text{Yes}}\right)$$

$$= 1/6 \times 2/6 \times 1 = 2/36 = 1/18$$

$$P(\text{No}/X) = P\left(\frac{\text{Single Parent}}{\text{No}}\right) * P\left(\frac{\text{Young}}{\text{No}}\right) * P\left(\frac{\text{High}}{\text{No}}\right)$$

$$= 1/4 \times 1/4 \times 0$$

$$= 0$$

∴ Final Probabilities are : ~~1/3 / 1/3 + 0~~

$$P(\text{Yes}/X) = \frac{1/18}{1/18 + 0} = 1$$

$$P(\text{No}/X) = \frac{0}{1/18 + 0} = 0$$

∴ For the given conditions X they "WILL BUY A CAR"

Q5]

```
S(tokens):
        NP(tokens)
        VP(tokens)
```

```
NP(tokens):
        DP(tokens)
```

```
DP(tokens):
        Det(tokens)
        NN(tokens)
```

```
Det(tokens):
        if tokens[0] == 'a' or tokens[0] == 'the':
            tokens.pop(0)

        else:
            raise ValueError("Error, got" + tokens[0])
```

```
NN(tokens):
        if tokens[0] == 'car' or tokens[0] == 'bike':
            tokens.pop(0)

        else:
            raise ValueError("Error, got" + tokens[0])
```

CLASSMATE

```
def VBR (tokens):
    if tokens[0] == 'running' or tokens[0] == 'stopping'
        tokens.pop (0)
    else:
        raise ValueError ("Expected running or stopping,
                          but got " + tokens[0])
```