# Implement Hidden Markov Model to predict the Part of the Speech Tagging:

**Tool**: Python

**Libraries Used**: pandas, tabulate

**Sample Problem**: Implement a POS Tagger using Hidden Markov Models

**Input Type:** Implement Observed States, Hidden States, Emission Probabilities, Transition Probabilities, Input Probabilities

**Output Type**: Sequences and Probability Scores

**Implementation:**

import pandas as pd

from tabulate import tabulate

# given states - what are the possible combinations

# total number of combinations is (number of possible states)^(sequence length)

def generate\_sequence(states, sequence\_length):

all\_sequences = []

nodes = []

depth = sequence\_length

def gen\_seq\_recur(states, nodes, depth):

if depth == 0:

#print nodes

all\_sequences.append(nodes)

else:

for state in states:

temp\_nodes = list(nodes)

temp\_nodes.append(state)

gen\_seq\_recur(states,temp\_nodes,depth-1)

gen\_seq\_recur(states,[],depth)

return all\_sequences

def score\_sequences(sequences,initial\_probs,transition\_probs,emission\_probs,obs):

best\_score = -1

best\_sequence = None

sequence\_scores = []

for seq in sequences:

total\_score = 1

first = True

for i in range(len(seq)):

state\_score = 1

# compute transitition probability score

if first == True:

state\_score \*= initial\_probs[seq[i]]

# reset first flag

first = False

else:

state\_score \*= transition\_probs[seq[i] + "|" + seq[i-1]]

# add to emission probability score

state\_score \*= emission\_probs[obs[i] + "|" + seq[i]]

# update the total score

#print state\_score

total\_score \*= state\_score

sequence\_scores.append(total\_score)

return sequence\_scores

# pretty printing our distributions

def pretty\_print\_probs(distribs):

print(distribs)

rows = set()

cols = set()

for val in distribs.keys():

temp = val.split("|")

rows.add(temp[0])

cols.add(temp[1])

rows = list(rows)

cols = list(cols)

df = []

for i in range(len(rows)):

temp = []

for j in range(len(cols)):

temp.append(distribs[rows[i]+"|"+cols[j]])

df.append(temp)

I = pd.Index(rows, name="rows")

C = pd.Index(cols, name="cols")

df = pd.DataFrame(data=df,index=I, columns=C)

print (tabulate(df, headers='keys', tablefmt='psql'))

def initializeSequences(\_obs):

# Generate list of sequences

seqLen = len(\_obs)

seqs = generate\_sequence(states, seqLen)

# Score sequencessc

seq\_scores = score\_sequences(seqs,initial\_probs,transition\_probs,emission\_probs,obs)

return (seqLen,seqs,seq\_scores)

states = ['Noun','Verb','Determiner']

initial\_probs = {'Noun':0.9,'Verb':0.05,'Determiner':0.05}

transition\_probs = {'Noun|Noun':0.1,'Noun|Verb':0.1,'Noun|Determiner':0.8,

'Verb|Noun':0.8,'Verb|Verb':0.1,'Verb|Determiner':0.1,

'Determiner|Noun':0.1,'Determiner|Verb':0.8,'Determiner|Determiner':0.1}

emission\_probs = {'Vimal|Noun':0.9,'taught|Noun':0.05,'the|Noun':0.05,'class|Noun':0.9,\

'Vimal|Verb':0.05,'taught|Verb':0.9,'the|Verb':0.05,'class|Verb':0.05,\

'Vimal|Determiner':0.05,'taught|Determiner':0.05,'the|Determiner':0.9,'class|Determiner':0.05}

print("Initial Distributions")

print(initial\_probs)

print("\nTransition Probabilities")

pretty\_print\_probs(transition\_probs)

print("\nEmission Probabilities")

pretty\_print\_probs(emission\_probs)

obs = ['Vimal','taught','the','class']

# print results

print("\nScores")

# Generate list of sequences

sequence\_length,sequences,sequence\_scores = initializeSequences(obs)

# Display sequence scores

for i in range(len(sequences)):

print("Sequence:%-60s Score:%0.6f" % (sequences[i],sequence\_scores[i]))

# Display the winning score

print("\n Best Sequence")

print(sequences[sequence\_scores.index(max(sequence\_scores))],max(sequence\_scores))

**Output:**

Scores

Sequence:['Noun', 'Noun', 'Noun', 'Noun'] Score:0.000002

Sequence:['Noun', 'Noun', 'Noun', 'Verb'] Score:0.000001

Sequence:['Noun', 'Noun', 'Noun', 'Determiner'] Score:0.000000

Sequence:['Noun', 'Noun', 'Verb', 'Noun'] Score:0.000015

Sequence:['Noun', 'Noun', 'Verb', 'Verb'] Score:0.000001

Sequence:['Noun', 'Noun', 'Verb', 'Determiner'] Score:0.000006

Sequence:['Noun', 'Noun', 'Determiner', 'Noun'] Score:0.000262

Sequence:['Noun', 'Noun', 'Determiner', 'Verb'] Score:0.000002

Sequence:['Noun', 'Noun', 'Determiner', 'Determiner'] Score:0.000002

Sequence:['Noun', 'Verb', 'Noun', 'Noun'] Score:0.000262

Sequence:['Noun', 'Verb', 'Noun', 'Verb'] Score:0.000117

Sequence:['Noun', 'Verb', 'Noun', 'Determiner'] Score:0.000015

Sequence:['Noun', 'Verb', 'Verb', 'Noun'] Score:0.000262

Sequence:['Noun', 'Verb', 'Verb', 'Verb'] Score:0.000015

Sequence:['Noun', 'Verb', 'Verb', 'Determiner'] Score:0.000117

Sequence:['Noun', 'Verb', 'Determiner', 'Noun'] Score:0.302331

Sequence:['Noun', 'Verb', 'Determiner', 'Verb'] Score:0.002100

Sequence:['Noun', 'Verb', 'Determiner', 'Determiner'] Score:0.002100

Sequence:['Noun', 'Determiner', 'Noun', 'Noun'] Score:0.000015

Sequence:['Noun', 'Determiner', 'Noun', 'Verb'] Score:0.000006

Sequence:['Noun', 'Determiner', 'Noun', 'Determiner'] Score:0.000001

Sequence:['Noun', 'Determiner', 'Verb', 'Noun'] Score:0.000002

Sequence:['Noun', 'Determiner', 'Verb', 'Verb'] Score:0.000000

Sequence:['Noun', 'Determiner', 'Verb', 'Determiner'] Score:0.000001

Sequence:['Noun', 'Determiner', 'Determiner', 'Noun'] Score:0.000262

Sequence:['Noun', 'Determiner', 'Determiner', 'Verb'] Score:0.000002

Sequence:['Noun', 'Determiner', 'Determiner', 'Determiner'] Score:0.000002

Sequence:['Verb', 'Noun', 'Noun', 'Noun'] Score:0.000000

Sequence:['Verb', 'Noun', 'Noun', 'Verb'] Score:0.000000

Sequence:['Verb', 'Noun', 'Noun', 'Determiner'] Score:0.000000

Sequence:['Verb', 'Noun', 'Verb', 'Noun'] Score:0.000000

Sequence:['Verb', 'Noun', 'Verb', 'Verb'] Score:0.000000

Sequence:['Verb', 'Noun', 'Verb', 'Determiner'] Score:0.000000

Sequence:['Verb', 'Noun', 'Determiner', 'Noun'] Score:0.000001

Sequence:['Verb', 'Noun', 'Determiner', 'Verb'] Score:0.000000

Sequence:['Verb', 'Noun', 'Determiner', 'Determiner'] Score:0.000000

Sequence:['Verb', 'Verb', 'Noun', 'Noun'] Score:0.000000

Sequence:['Verb', 'Verb', 'Noun', 'Verb'] Score:0.000000

Sequence:['Verb', 'Verb', 'Noun', 'Determiner'] Score:0.000000

Sequence:['Verb', 'Verb', 'Verb', 'Noun'] Score:0.000000

Sequence:['Verb', 'Verb', 'Verb', 'Verb'] Score:0.000000

Sequence:['Verb', 'Verb', 'Verb', 'Determiner'] Score:0.000000

Sequence:['Verb', 'Verb', 'Determiner', 'Noun'] Score:0.000117

Sequence:['Verb', 'Verb', 'Determiner', 'Verb'] Score:0.000001

Sequence:['Verb', 'Verb', 'Determiner', 'Determiner'] Score:0.000001

Sequence:['Verb', 'Determiner', 'Noun', 'Noun'] Score:0.000000

Sequence:['Verb', 'Determiner', 'Noun', 'Verb'] Score:0.000000

Sequence:['Verb', 'Determiner', 'Noun', 'Determiner'] Score:0.000000

Sequence:['Verb', 'Determiner', 'Verb', 'Noun'] Score:0.000000

Sequence:['Verb', 'Determiner', 'Verb', 'Verb'] Score:0.000000

Sequence:['Verb', 'Determiner', 'Verb', 'Determiner'] Score:0.000000

Sequence:['Verb', 'Determiner', 'Determiner', 'Noun'] Score:0.000006

Sequence:['Verb', 'Determiner', 'Determiner', 'Verb'] Score:0.000000

Sequence:['Verb', 'Determiner', 'Determiner', 'Determiner'] Score:0.000000

Sequence:['Determiner', 'Noun', 'Noun', 'Noun'] Score:0.000000

Sequence:['Determiner', 'Noun', 'Noun', 'Verb'] Score:0.000000

Sequence:['Determiner', 'Noun', 'Noun', 'Determiner'] Score:0.000000

Sequence:['Determiner', 'Noun', 'Verb', 'Noun'] Score:0.000000

Sequence:['Determiner', 'Noun', 'Verb', 'Verb'] Score:0.000000

Sequence:['Determiner', 'Noun', 'Verb', 'Determiner'] Score:0.000000

Sequence:['Determiner', 'Noun', 'Determiner', 'Noun'] Score:0.000006

Sequence:['Determiner', 'Noun', 'Determiner', 'Verb'] Score:0.000000

Sequence:['Determiner', 'Noun', 'Determiner', 'Determiner'] Score:0.000000

Sequence:['Determiner', 'Verb', 'Noun', 'Noun'] Score:0.000000

Sequence:['Determiner', 'Verb', 'Noun', 'Verb'] Score:0.000000

Sequence:['Determiner', 'Verb', 'Noun', 'Determiner'] Score:0.000000

Sequence:['Determiner', 'Verb', 'Verb', 'Noun'] Score:0.000000

Sequence:['Determiner', 'Verb', 'Verb', 'Verb'] Score:0.000000

Sequence:['Determiner', 'Verb', 'Verb', 'Determiner'] Score:0.000000

Sequence:['Determiner', 'Verb', 'Determiner', 'Noun'] Score:0.000117

Sequence:['Determiner', 'Verb', 'Determiner', 'Verb'] Score:0.000001

Sequence:['Determiner', 'Verb', 'Determiner', 'Determiner'] Score:0.000001

Sequence:['Determiner', 'Determiner', 'Noun', 'Noun'] Score:0.000000

Sequence:['Determiner', 'Determiner', 'Noun', 'Verb'] Score:0.000000

Sequence:['Determiner', 'Determiner', 'Noun', 'Determiner'] Score:0.000000

Sequence:['Determiner', 'Determiner', 'Verb', 'Noun'] Score:0.000000

Sequence:['Determiner', 'Determiner', 'Verb', 'Verb'] Score:0.000000

Sequence:['Determiner', 'Determiner', 'Verb', 'Determiner'] Score:0.000000

Sequence:['Determiner', 'Determiner', 'Determiner', 'Noun'] Score:0.000001

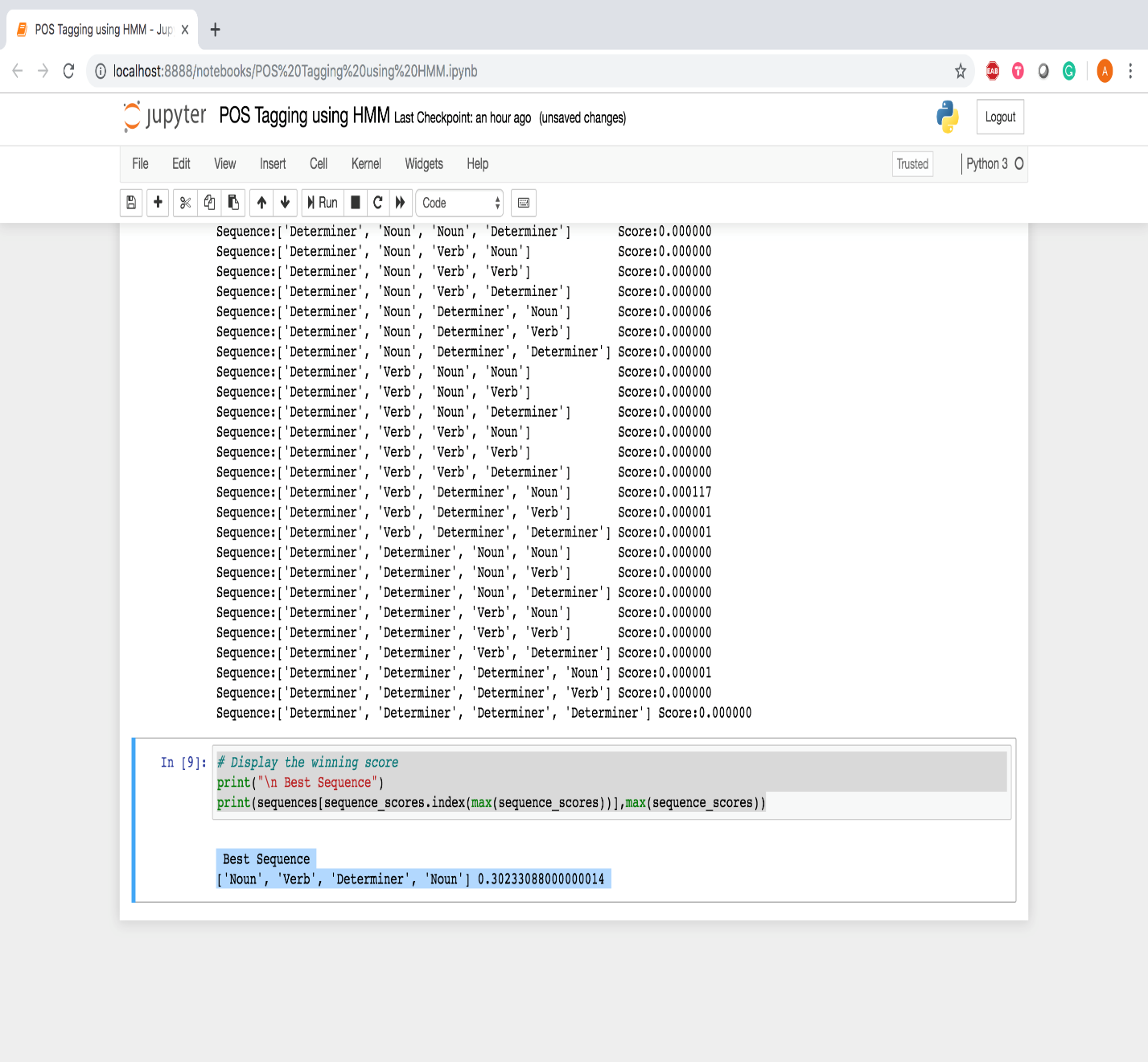
Sequence:['Determiner', 'Determiner', 'Determiner', 'Verb'] Score:0.000000

Sequence:['Determiner', 'Determiner', 'Determiner', 'Determiner'] Score:0.000000

Best Sequence

['Noun', 'Verb', 'Determiner', 'Noun'] 0.30233088000000014

**Screenshot**

****

**Lab Exercises**

* Construct the initial , transition and emission probabilities for below corpus of sentences
* Boys are taller.
* This is the tree.
* She is a tall girl.
* Trees are more.
* Girls are more than boys.
* The tall tree is falling.
* Using the above model Predict the PoS tag for the sentence “**Girls are falling**”