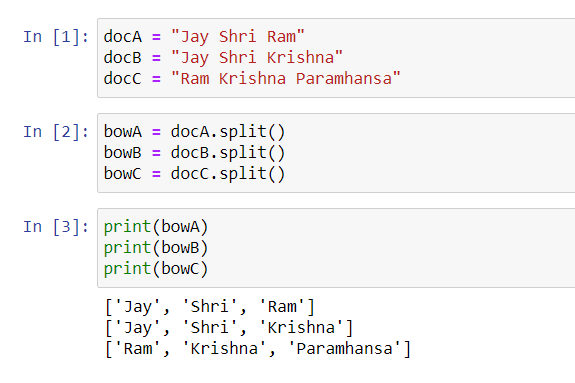
**EXPERIMENT-11**

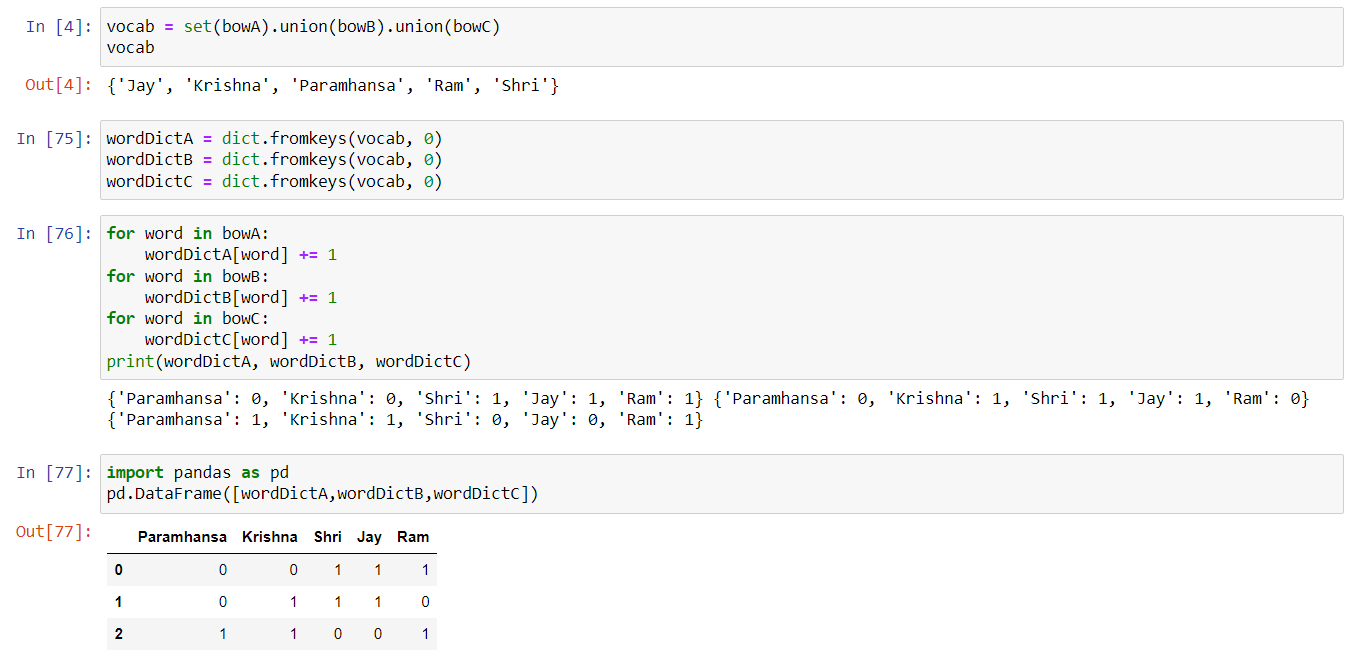
**AIM:** *Program to calculate TF/IDF , feature extraction and finding unique words*

**CODE**

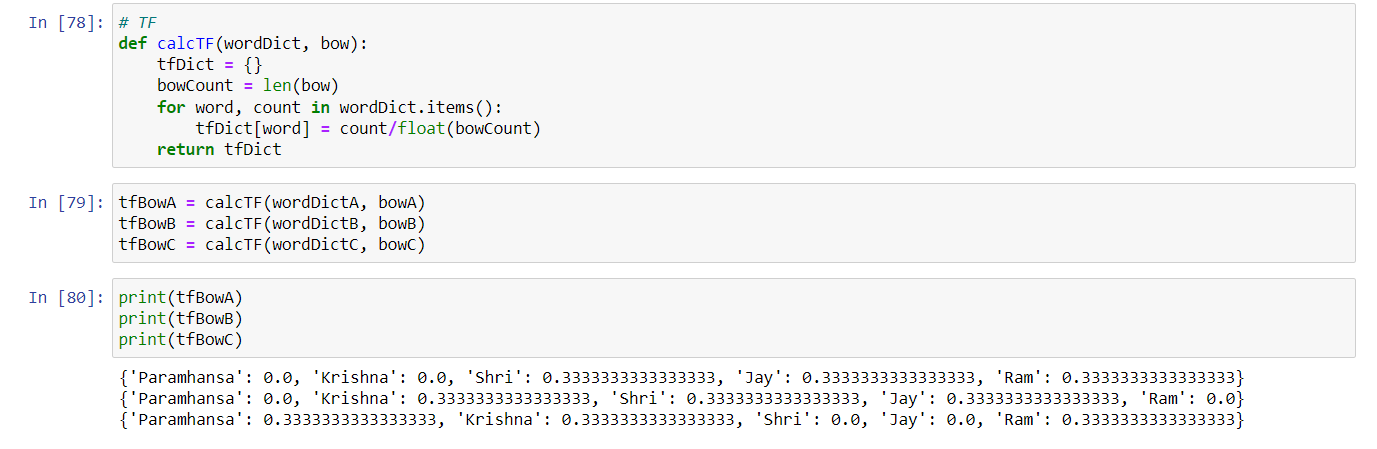
# Feature Extraction

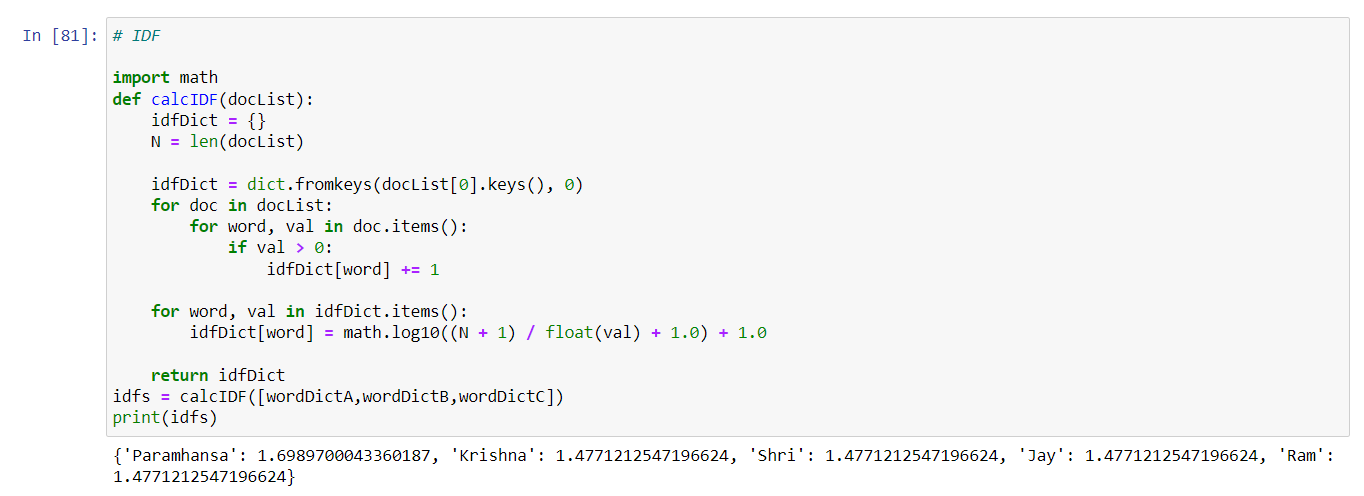


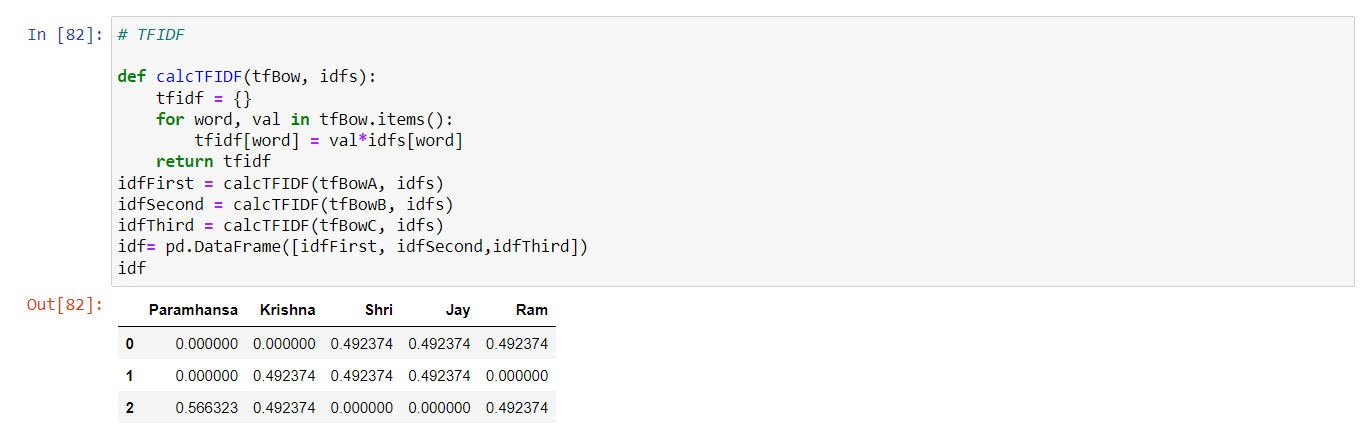
**Finding Unique Words**



# TF-IDF







**EXPERIMENT-12**

**AIM:** *Program for data collection , remove stop words ,data generation , tensor flow graph and word vector .*

**CODE**

# Data Collection

corpus = ['I like apple juice',

'I like orange juice',

'king is a strong man',

'queen is a wise woman',

'boy is a young man',

'girl is a young woman',

'prince is a young king',

'princess is a young queen',

'man is strong',

'woman is pretty',

'prince is a boy will be king',

'princess is a girl will be queen',

'Apple is good place for work']

Corpus

**OUTPUT**

['I like apple juice',

'I like orange juice',

'king is a strong man',

'queen is a wise woman',

'boy is a young man',

'girl is a young woman',

'prince is a young king',

'princess is a young queen',

'man is strong',

'woman is pretty',

'prince is a boy will be king',

'princess is a girl will be queen',

'Apple is good place for work']

# Remove stop words

def remove\_stop\_words(corpus):

stop\_words = ['is', 'a', 'will', 'be']

results = []

for text in corpus:

tmp = text.split(' ')

for stop\_word in stop\_words:

if stop\_word in tmp:

tmp.remove(stop\_word)

results.append(" ".join(tmp))

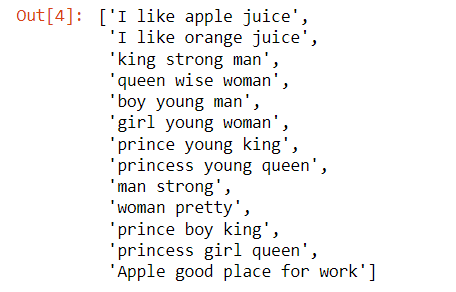
return results

#After removing all stop-words

corpus = remove\_stop\_words(corpus)

corpus

**OUTPUT**



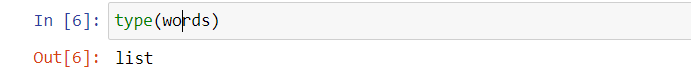
words = []

for text in corpus:

for word in text.split(' '):

words.append(word)

type(words)





# Data Generation

word2int = {}

#Here we assisgned number to each word store it into Dictionary

for i,word in enumerate(words):

word2int[word] = i

# Here we split corpus into sentences

sentences = []

for sentence in corpus:

sentences.append(sentence.split())

WINDOW\_SIZE = 2 # Dimension is 2 means "we consider 2 words from left and right to the centre word

data = []

for sentence in sentences:

for idx, word in enumerate(sentence):

for neighbor in sentence[max(idx - WINDOW\_SIZE, 0) : min(idx + WINDOW\_SIZE, len(sentence)) + 1] :

if neighbor != word:

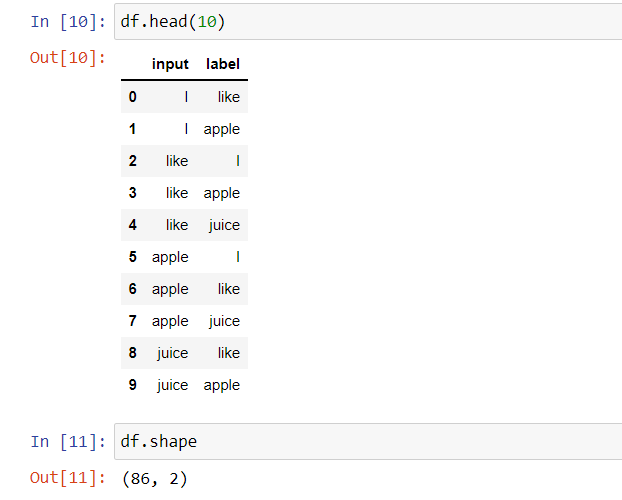
data.append([word, neighbor])

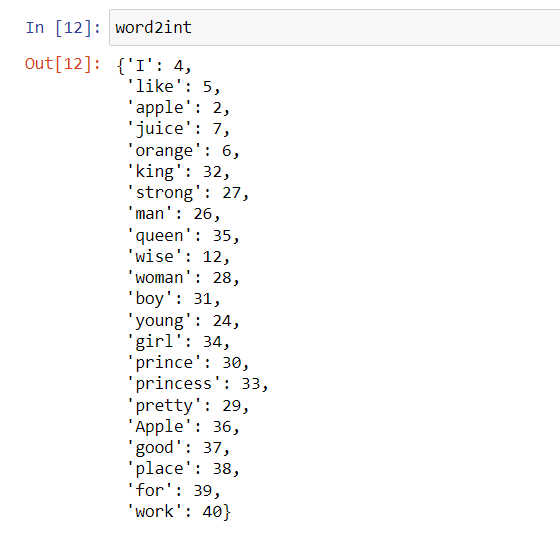
import pandas as pd

df = pd.DataFrame(data, columns = ['input', 'label'])

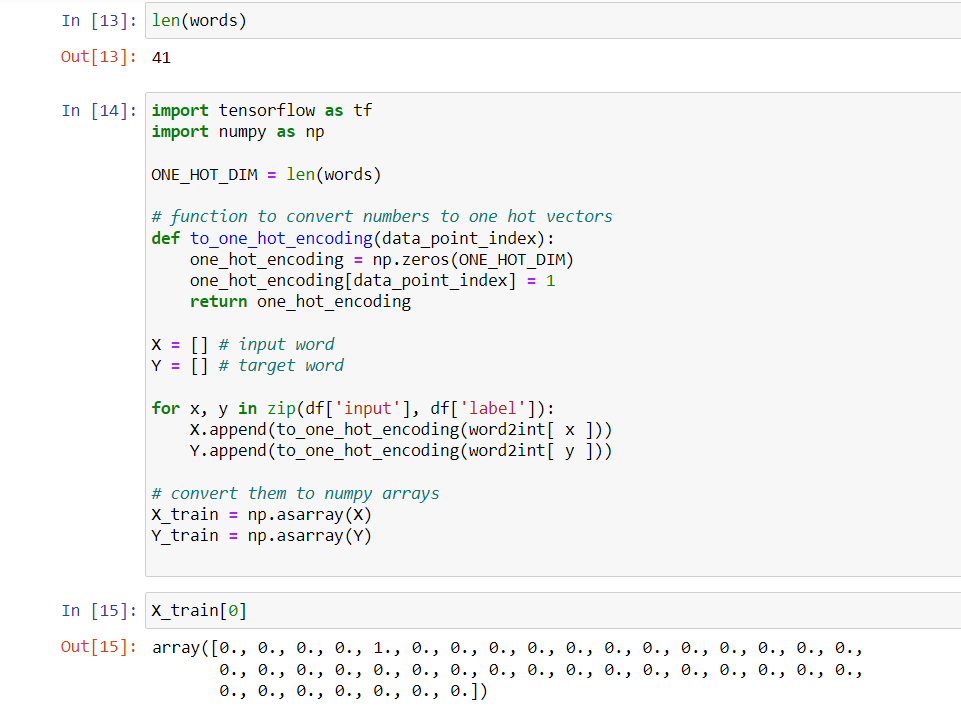
df



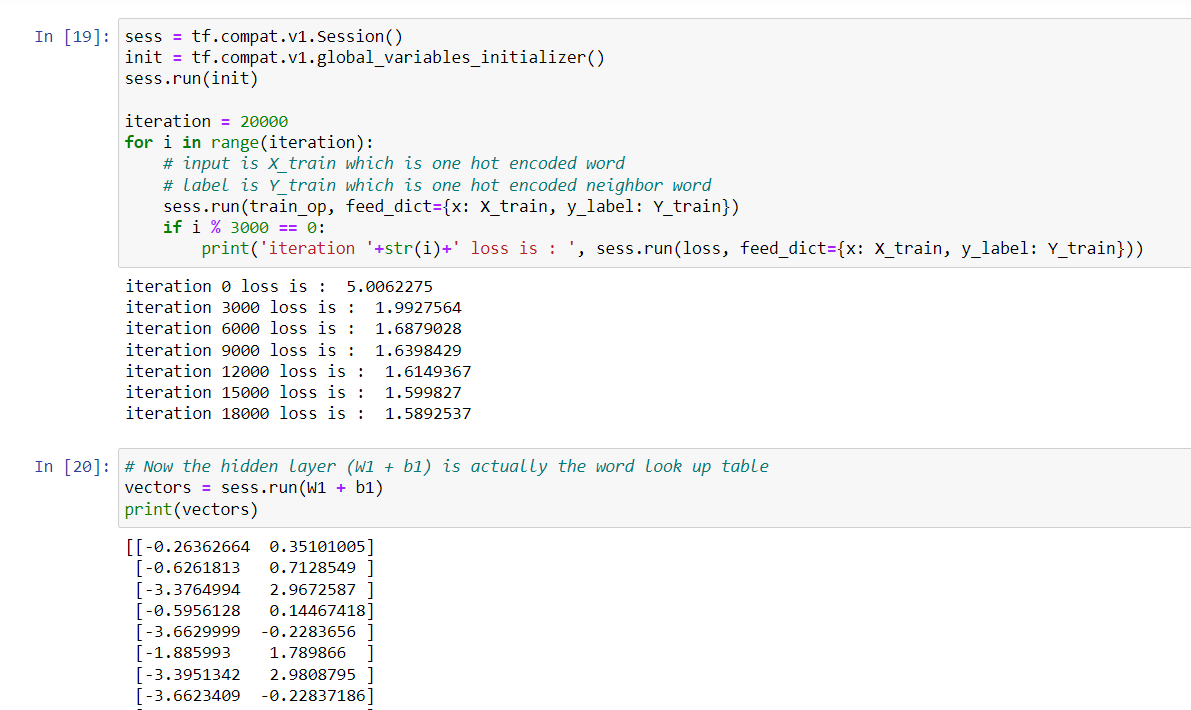


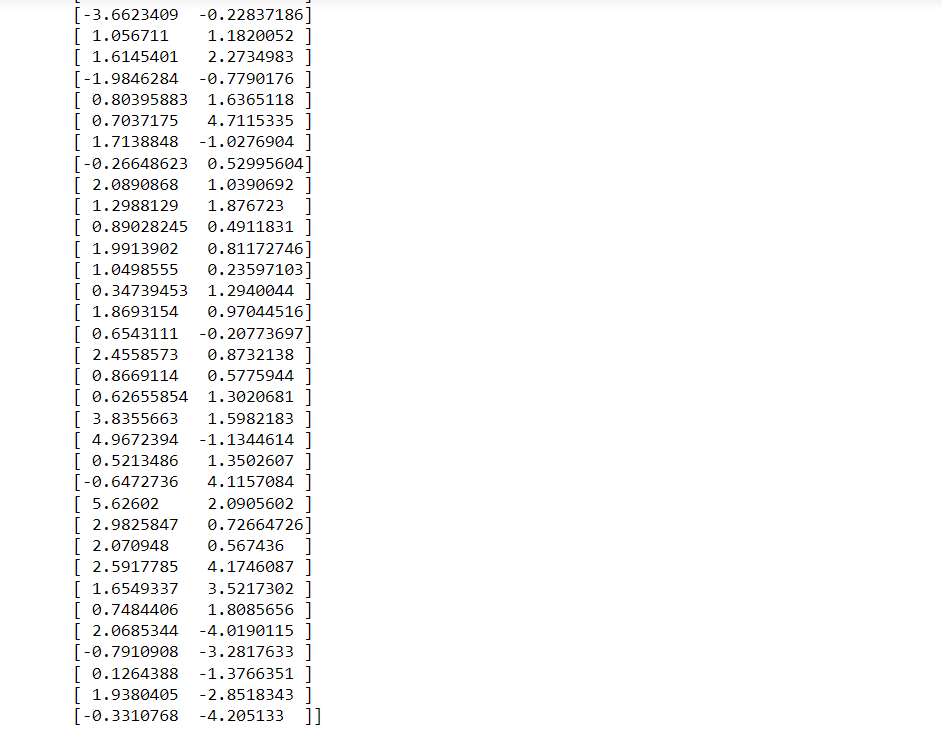
****

# Define Tensor flow Graph

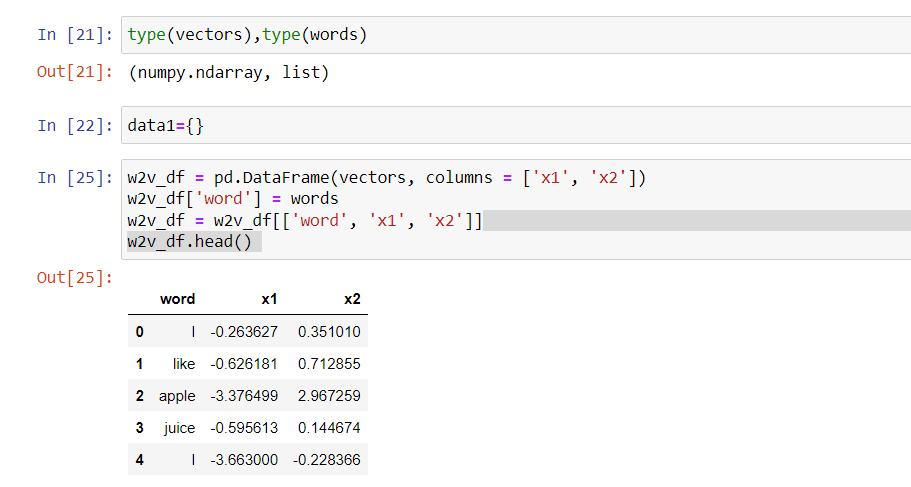


# Train

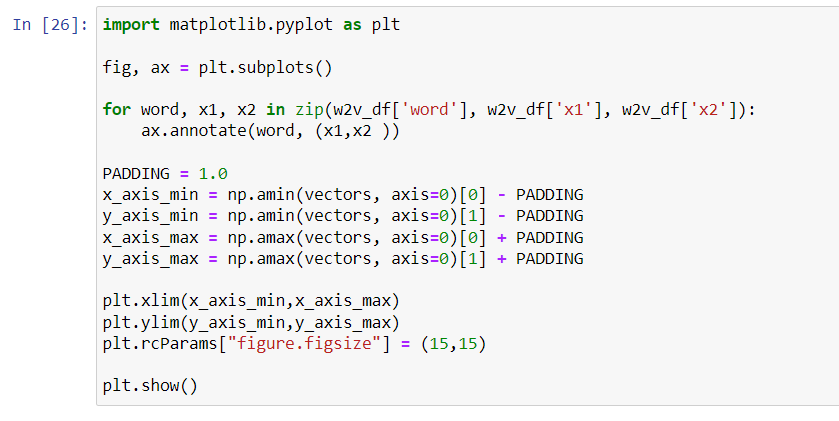




# Word vector in table



# Word vector in 2d chart



Output:

