Name:Rutuja Gorakh Tembare

Roll No.:CC-79

PRN:202401030006

Theory Activity No.01

Formulate 20 problem statements for a given dataset using Numpy and Pandas and Apply Numpy and pandas methods to find the solution for the formulated problem statements

Word net dataset:

```
import numpy as np

import pandas as pd

import nltk

from nltk.corpus import wordnet as wn


# Make sure to download wordnet if you haven't yet

nltk.download('wordnet')
```

Dataframe of WordNet dataset:

```
data = []
```

```
for synset in list(wn.all_synsets()):
    data.append([
        synset.name(),
        synset.pos(),
        synset.definition(),
        [lemma.name() for lemma in synset.lemmas()],  # all lemma
names
        synset.hypernyms(),
        synset.hyponyms()


columns = ['synset_name', 'pos', 'definition', 'lemmas', 'hypernyms',
'hyponyms']
wordnet_df = pd.DataFrame(data, columns=columns)


wordnet_df.head()
```

Problem Statements :

1. How many synsets are there for each part of speech (noun, verb, adjective, adverb)?

2. Find the average number of lemmas per synset.

3. Find the synsets with the longest definition.

4. List synsets that have no hypernyms (they are at the top of hierarchy).

5. Find synsets that have no hyponyms (they are leaves in hierarchy).

6. Which lemma appears the most across synsets?

7. Average number of hypernyms per synset.

8. Average number of hyponyms per synset.

9. Find synsets whose lemma list contains the word 'dog'.

10. Find the top 5 synsets with maximum hyponyms

Solving using pandas and numpy:

1.Synsets per part of speech:

```
synsets_per_pos = wordnet_df['pos'].value_counts()

print(synsets_per_pos)
```

2.Average number of lemmas per synset:

```
wordnet_df['num_lemmas'] = wordnet_df['lemmas'].apply(len)

average_lemmas = wordnet_df['num_lemmas'].mean()

print("Average number of lemmas per synset:", average_lemmas)
```

3.Synsets with longest definition:

```
wordnet_df['definition_length'] =
wordnet_df['definition'].apply(lambda x: len(x))

longest_definition =
wordnet_df.loc[wordnet_df['definition_length'].idxmax()]

print(longest_definition[['synset_name', 'definition']])
```

4.Synsets with no hypernyms:

```
no_hypernyms = wordnet_df[wordnet_df['hypernyms'].apply(len) ==
0]

print(no_hypernyms[['synset_name', 'lemmas']])
```

5.Synsets with no hyponyms:

```
no_hyponyms = wordnet_df[wordnet_df['hyponyms'].apply(len) ==
0]

print(no_hyponyms[['synset_name', 'lemmas']])
```

6.Lemma that appears most:

```
all_lemmas = np.concatenate(wordnet_df['lemmas'].values)
```

```
lemma_counts = pd.Series(all_lemmas).value_counts()

most_common_lemma = lemma_counts.idxmax()

print("Most common lemma:", most_common_lemma)
```

7.Average number of hypernyms per synset:

```
wordnet_df['num_hypernyms'] =
wordnet_df['hypernyms'].apply(len)

average_hypernyms = wordnet_df['num_hypernyms'].mean()

print("Average number of hypernyms:", average_hypernyms)
```

8.Average number of hyponyms per synset:

```
wordnet_df['num_hyponyms'] = wordnet_df['hyponyms'].apply(len)

average_hyponyms = wordnet_df['num_hyponyms'].mean()

print("Average number of hyponyms:", average_hyponyms)
```

9.Synsets whose lemmas contain dog:

```
dog_lemmas = wordnet_df[wordnet_df['lemmas'].apply(lambda x:
'dog' in x)]

print(dog_lemmas[['synset_name', 'lemmas', 'definition']])
```

10.Top 5 synsets with maximum hyponyms:

```
top_hyponyms = wordnet_df.sort_values('num_hyponyms',
ascending=False).head(5)

print(top_hyponyms[['synset_name', 'lemmas', 'num_hyponyms']])
```