```
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
import nltk
import pandas as pd
import os
```

```
os.listdir('/content/Dataset')
```

```
['train.csv', 'clean_train.csv', 'test.csv']
```

```
df = pd.read_csv('./Dataset/clean_train.csv')
df.head(5)
```

	Unnamed: 0	id	movie_name	synopsis	genre
0	0	44978	Super Me	A young scriptwriter starts bringing valuable ...	fantasy
1	1	50185	Entity Project	A director and her friends renting a haunted h...	horror
2	2	34131	Behavioral Family Therapy for Serious Psychiat...	This is an educational video for families and ...	family
3	3	78522	Blood Glacier	Scientists working in the Austrian Alps discov...	scifi
4	4	2206	Apat na anino	Buy Day - Four Men Widely - Apart in Life - By...	action

Next steps: Generate code with df View recommended plots New interactive sheet

```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
stop_words = set(nltk.corpus.stopwords.words('english'))
stop_words
```

```
 's',
 'same',
 'shan',
 "shan't",
 'she',
 "she's",
 'should',
 "should've",
 'shouldn',
 "shouldn't",
 'so'
```

```
def remove_stopwords(text:str):
    words = text.split()
    filtered_words = [word for word in words if word.lower() not in stop_words]
    return ' '.join(filtered_words)
```

```
df['filtered_synopsis'] = df['synopsis'].apply(remove_stopwords)
df['filtered_synopsis'][:5]
```

```
                                   filtered_synopsis
0     young scriptwriter starts bringing valuable ob...
1    director friends renting haunted house capture...
2     educational video families family therapists d...
3     Scientists working Austrian Alps discover glac...
4   Buy Day - Four Men Widely - Apart Life - Night...

dtype: object
```

```
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
True
```

```
lemmatizer = WordNetLemmatizer()
```

```
def lemmatize_words(text):
    words = text.split()
    lemmatized_words = [lemmatizer.lemmatize(word) for word in words]
    return ' '.join(lemmatized_words)
```

```
df['lemmatized_synopsis'] = df['filtered_synopsis'].apply(lemmatize_words)
df['lemmatized_synopsis'][:5]
```

```
                                 lemmatized_synopsis
0      young scriptwriter start bringing valuable obj...
1     director friend renting haunted house capture ...
2     educational video family family therapist desc...
3     Scientists working Austrian Alps discover glac...
4   Buy Day - Four Men Widely - Apart Life - Night...

dtype: object
```

```
df.to_csv('./Dataset/lemmatized_data.csv')
```