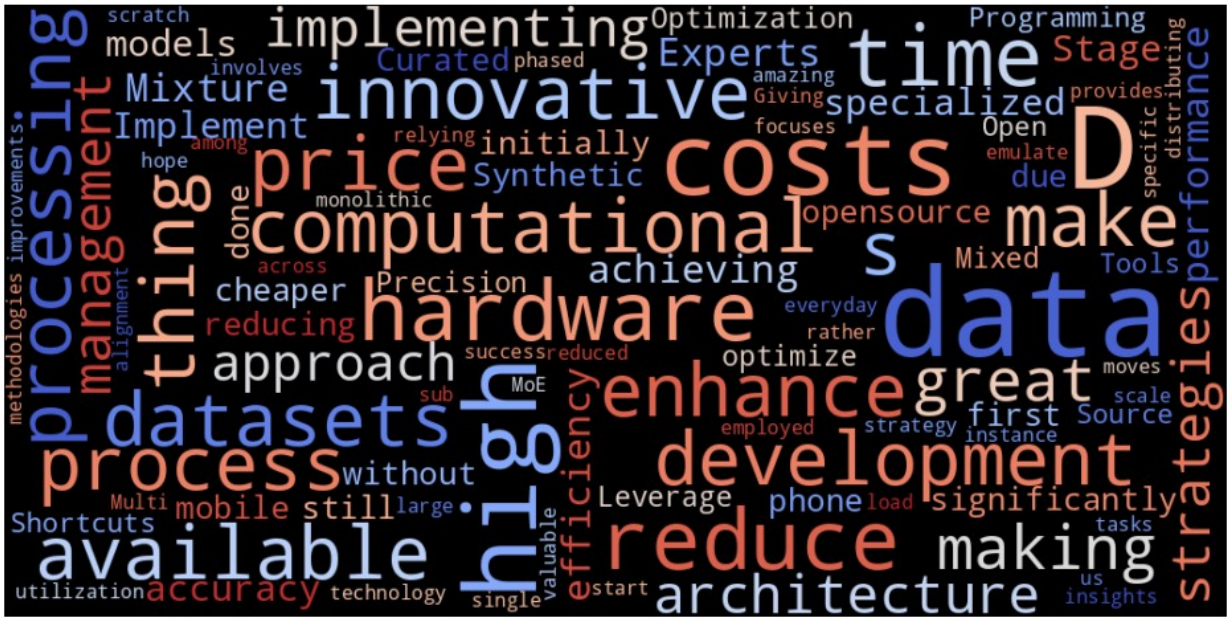
```
text="""
Training a large-scale AI model like amazing DeepSeek (Giving us hope
that technology is making great moves everyday) from scratch at a
reduced cost involves implementing innovative strategies across data
management, model architecture, hardware utilization, and training
methodologies. DeepSeek’s approach provides valuable insights into
achieving high performance with cost efficiency. Here’s how you can
emulate their success:

. Innovative Model Architecture
Mixture of Experts (MoE): DeepSeek employed a “mixture of experts”
approach, distributing tasks among specialized sub-models rather than
relying on a single, monolithic model. This strategy reduces
computational load and enhances efficiency.
Multi-Stage Training: Implement a phased training process, where each
stage focuses on specific improvements such as accuracy or alignment.
For instance, start with general text data pretraining, followed by
reinforcement learning on user feedback to enhance conversational
abilities.
Avoid R&D cost: Implementing something for the second time is
significantly cheaper since minimal research and development (R&D) is
required. R&D is typically the biggest expense in any innovation. Take
the invention of the light bulb or mobile phone as an example –
initially, they were sold at extremely high prices due to the
substantial R&D costs. However, over time, as development costs were
recovered, prices dropped, making them affordable for as little as $50
per mobile phone. Initially, selling them at such a low price was
impossible due to the high investment in R&D and as OPENAI is still
under their R&D process of achieving more great things, so they can
not still make them opensource or lower cost that much, as they have
their own R&D cost of doing things first time in the world. Once the
thing is done first time, there are many researches available done by
universities and institutes and are opensource, those can really
reduce the company cost of R&D.
2. Efficient Data Management
High-Quality Curated Datasets: Utilize well-curated datasets to ensure
the model learns from accurate and relevant information, reducing the
need for extensive data cleaning and processing.
Synthetic Data Generation: Generate synthetic data to augment training
datasets, especially in specialized domains, to enhance model
robustness without incurring high data acquisition costs.
3. Hardware Optimization
Leverage Available Hardware: DeepSeek capitalized on available high-
performance chips, such as Nvidia’s H800, to optimize their training
process. Assess the hardware at your disposal and optimize your
training algorithms to make the most of it.
Mixed-Precision Training: Implement mixed-precision arithmetic to
reduce memory usage and increase computational speed without
compromising model accuracy.
```

```
4. Cost-Effective Training Strategies
Programming Shortcuts: DeepSeek utilized innovative programming
shortcuts to reduce data-processing requirements, significantly
cutting down training costs. Explore algorithmic optimizations that
can streamline computations.
Open-Source Tools: Leverage open-source frameworks and tools to build
and train your models, minimizing software development costs.
To make training a model like DeepSeek even cheaper, you can focus on
reducing computational costs, optimizing data processing, and
leveraging alternative training approaches. Here are additional cost-
saving techniques:




"""

from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt

# Define additional stopwords

custom_stopwords = set(STOPWORDS).union({"DeepSeek", "OPENAI",
"model", "training", "AI", "cost"})

# Generate Word Cloud with improvements
wordcloud = WordCloud(
    width=1000, height=500,
    background_color="black",  # Better contrast
    colormap="coolwarm",        # Improved color palette
    stopwords=custom_stopwords, # Remove unimportant words
    contour_color='white',      # Adds an outline
    contour_width=1.5,
    max_words=100,              # Limit number of words
    max_font_size=100           # Improve font size
).generate(text)

# Display the Word Cloud
plt.figure(figsize=(12, 6))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")  # Hide axes
plt.title("Key Concepts in DeepSeek AI Model Optimization",
fontsize=14, color="white")
plt.show()
```




```
from google.colab import drive
import pandas as pd
from wordcloud import WordCloud
import matplotlib.pyplot as plt
import numpy as np
import re
import nltk
from nltk.corpus import stopwords

drive.mount("/content/drive")
```

```
Mounted at /content/drive
```

```
# Load the Excel file
file_path = "/content/drive/MyDrive/Course Work/Sem 4/Data Analysis and Visualization/Lab 6/Person_Data.xlsx"  # Update with the correct file path
df = pd.read_excel(file_path, sheet_name="Sheet1")

df
```

```
{"type":"dataframe","variable_name":"df"}
```

```
# Combine all attribute columns into a single text corpus
text_data = df.iloc[:, 2:].astype(str).values.flatten()  # Extract only attribute columns

cleaned_text_data = [word for word in text_data if not (isinstance(word, float) and np.isnan(word)) and word != "nan"]

cleaned_text_data
```

```
['Funny',
 'Mature',
 'Understanding',
 'Kind',
 'Short tempered',
 'Intellegent',
 'Hard working',
 'Ambitions',
 'Courageous',
 'Caring',
 'Emotionaally Distant',
 'Hard working',
 'Caring',
 'Progressive Mind',
 'Honest',
 'Holy Person',
 'Loyal',
 'Trustwarthy',
 'Self Confidance',
 'Emotion less',
 'Loving',
 'Helping',
 'Emotional',
 'Strong',
 'Short tempered',
 'Short Height',
 'Hard working',
 'Best Cook',
 'Caring',
 'Beautiful',
 'Professional ',
 'Casual in study',
 'Low Marks',
 'shortcut study',
 'Humorous',
 'standup comedian',
 'makes fun of everyone',
 'Helpful',
 'Voilent',
 'Kind',
 'Handsome',
 'Funny',
 'Intelligent',
 'Sportsman',
 'Slim',
 'Romantic',
 'Loyal',
 'Writer',
 'Honest',
 'Rich',
```

```
 'Caring',
 'Short',
 'thin',
 'tempered',
 'Hero',
 'funny',
 'Funny',
 'Smart',
 'Rich',
 'no emotions',
 'agrees to everything',
 'Supportive',
 'Understanding',
 'Compromising',
 'Good Memory',
 'Helping',
 'selfless',
 'simple',
 'relational',
 'reasonable',
 'Helping Nature',
 'Ignorance',
 'Short tempered',
 'Caring',
 'Foodie',
 'Reader',
 'Socialist',
 'Geedy about new thing',
 'Awareness',
 'Overcomig failure',
 'Male',
 '19 years old',
 '5.7 Height',
 'weight-62',
 '6 feet',
 '65 kg',
 'male',
 'Vijaywada AP',
 'Likes Cricket',
 'Skill: Play with Keyboard',
 'Reader',
 'Likes Movies',
 'Fav Mov: Inception',
 '5.4 feet',
 '65 kg',
 'Beautiful',
 'Early Riser',
 'Listen Song',
 'Disciplined',
 'Emotional',
```

```
'Very Arogent',
'5.7 height',
'Slim',
'21 Years Old',
'Color: Brown',
'Doing Engineering',
'eye color: Brown',
'Native Lang: Telgu',
'65 kg',
'6.2 Feet',
'80 Kg',
'Age: 36',
'Highly Competetive',
'Basketball Player',
'Sarcastic',
'Atheletic Figure',
'Calm and Composed',
'Age: 20',
'Male',
'5.7 Feet',
'63 Kg',
'Eye Color: Brown',
'Hair Color: Black',
'Skin Color: Brown',
'Hobbies: Football, Comics',
'Skill: Computer Language',
'Food: Biryani',
'Movies: 99 Songs',
'43 Years',
'Female',
'5.7 Feet',
'65 Kg',
'Govt Emp',
'Kind',
'Loving',
'Eye Color: Black',
'Hair: Brown',
'163.5 Cm',
'62 Kg',
'Skintone: Pale',
'H Color: Black',
'Eye: Dark Brown',
'40 Years',
'Glasses: True',
'Married',
'Working: True',
'Job: Asstt Professor',
'Earring: True',
'Bindi: true',
'Children: 2',
```

```
'Hair Type: Silky',
'Skin Type: Dry',
'Fav Color: Yellow',
'Fav Food: Upma',
'Arrogant',
'Short temper',
'Overconfident',
'Hyper',
'Helpful',
'Highly Intelligent',
'Intriguing',
'Effortless',
'Lazy',
'Chill',
'Communication',
'Bad at Sports',
'Creative',
'Thoughtful',
'caring',
'cute',
'childish',
'Intelligent',
'Pretty Eyes',
'Handwriting',
'Loving',
'caring',
'Funny',
'friendly',
'curly hair',
'always panics',
'big eyes',
'smart',
'hardworking',
'multitasking',
'very emotional',
'shorter',
'Sweet',
'small',
'naughty',
'cute',
'mischievious',
'happy',
'sister',
'girl',
'intelligent',
'painter',
'coward',
'loving',
'caring',
'foodie',
```

```
'cry often',
'pretty',
'Short',
'Short hair',
'confident',
'kind',
'humour',
'lame',
'sports',
'lovable',
'short',
'fair',
'black eyes',
'confident',
'bold',
'childish',
'movies',
'chill',
'mature',
'Kind',
'good character',
'teddy bear',
'funny',
'Politician',
'calm',
'understanding',
'Occupation: Jila Sanyojak',
'Small hair',
'white skin',
'big face',
'lazy',
'fast thinking',
'helping nature',
'cricket',
'badminton',
'introvert',
'singing',
'weak communication',
'sport',
'driver',
'Good cook',
'beutiful',
'good dressing',
'humour',
'intelligent',
'decision making',
'emotional',
'fearless',
'strong',
'hairstylist',
```

```
 'knowledge',
 'singer',
 'sketcher',
 'introvert',
 'Beautiful',
 'loving',
 'caring',
 'hardworking',
 'support',
 'patience',
 'sacrifice',
 'helping',
 'strength',
 'strong',
 'motivator',
 'inspiring',
 'best cook',
 'friend',
 'short',
 'fast',
 'old',
 'legendary',
 'famous',
 'multi billonaire',
 'innovative',
 'male',
 'american',
 'english',
 'business',
 'space exploration',
 'lazy',
 'sleepy',
 'flute',
 'athletic',
 'bhajan',
 'polite ',
 'honest',
 'teacher',
 'overthinker',
 'puntual',
 'newspaper',
 'hard working',
 'kind',
 'physicque',
 'orthodox',
 'disciplined',
 'determined',
 'jolly',
 'ignorant',
 'eye glasses',
```

```
'gym',
'serious',
'Attractive',
'bold',
'strong',
'destructive',
'consistent',
'hard working',
'calm',
'hypertrophy',
'non controversial',
'mental',
'natural',
'fit',
'socially dead',
'COnfident',
'Helping',
'humble',
'strict',
'hardworking',
'honest',
'disciplined',
'tall',
'Introvert',
'helpful',
'conservative',
'disciplined',
'calm',
'humble',
'kind',
'honest',
'disciplined',
'tall',
'hardworking',
'Story Writer',
'animated show',
'manga comic',
'pirate',
'netflix',
'japanese',
'50 years',
'55 years',
'good character',
'male',
'brown',
'workout',
'hardworking',
'enjoyable',
'angry',
'social',
```

```
'non vegetarian ',
'humour',
'hardworking',
'happy',
'cool',
'social',
'non vegetarian ',
'humour',
'hardworking',
'enjoyable',
'cool',
'social',
'non vegetarian ',
'humour',
'video game',
'sleepy',
'artistic',
'thin',
'fair',
'sports',
'drive',
'introvert',
'shy',
'animal lover',
'anime',
'friendly',
'singer',
'coder',
'piano',
'reading',
'writing',
'best friend',
'bulky body',
'gym',
'funny',
'cricket',
'table tennis',
'caring',
'expressive',
'traveling',
'gentle',
'bargaining',
'Height 1.79 meter',
'74 kg',
'41 years',
'brown',
'englandactor',
'brown',
'leo',
'white',
```

```
 'height 183 cm',
 '95 kg',
 '18 years',
 'joyful',
 'short tempered',
 'badminton',
 'consistent',
 'cricket',
 'tallest',
 'space enthusiast',
 'friendly',
 'vizag',
 'dark',
 '19 years',
 'student']
```

```
nltk.download("stopwords")
```

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

```
True
```

```
stop_words = set(stopwords.words("english"))

extra_words = {
    "years", "old", "height", "meter", "kg", "cm", "color", "eye",
"hair",
    "male", "female", "working", "job", "govt", "assistant",
"professor",
    "true", "false", "children", "brown", "black", "white", "pale",
"dark",
    "student", "person", "occupation", "weight", "actor", "leo",
"english",
    "telugu", "japanese", "american", "native", "language", "study"
}

def clean_word_list(words):
    cleaned_words = set()  # Use a set to remove duplicates

    for word in words:
        word = word.lower().strip()  # Convert to lowercase and remove
extra spaces
        word = re.sub(r"[^a-z\s]", "", word)  # Remove numbers and
special characters
        word = re.sub(r"\s+", " ", word).strip()  # Remove extra
spaces

        if word and word not in stop_words and word not in
extra_words:
            cleaned_words.add(word)  # Add to set (removes duplicates)
```

```
    return sorted(list(cleaned_words))

cleaned_data = clean_word_list(cleaned_text_data)

cleaned_data
```

```
['age',
 'agrees to everything',
 'always panics',
 'ambitions',
 'angry',
 'animal lover',
 'animated show',
 'anime',
 'arrogant',
 'artistic',
 'atheletic figure',
 'athletic',
 'attractive',
 'awareness',
 'bad at sports',
 'badminton',
 'bargaining',
 'basketball player',
 'beautiful',
 'best cook',
 'best friend',
 'beutiful',
 'bhajan',
 'big eyes',
 'big face',
 'bindi true',
 'black eyes',
 'bold',
 'bulky body',
 'business',
 'calm',
 'calm and composed',
 'caring',
 'casual in study',
 'childish',
 'chill',
 'coder',
 'color brown',
 'communication',
 'compromising',
 'confident',
 'conservative',
 'consistent',
```

```
 'cool',
 'courageous',
 'coward',
 'creative',
 'cricket',
 'cry often',
 'curly hair',
 'cute',
 'decision making',
 'destructive',
 'determined',
 'disciplined',
 'doing engineering',
 'drive',
 'driver',
 'early riser',
 'earring true',
 'effortless',
 'emotion less',
 'emotionaally distant',
 'emotional',
 'englandactor',
 'enjoyable',
 'expressive',
 'eye color black',
 'eye color brown',
 'eye dark brown',
 'eye glasses',
 'fair',
 'famous',
 'fast',
 'fast thinking',
 'fav color yellow',
 'fav food upma',
 'fav mov inception',
 'fearless',
 'feet',
 'fit',
 'flute',
 'food biryani',
 'foodie',
 'friend',
 'friendly',
 'funny',
 'geedy about new thing',
 'gentle',
 'girl',
 'glasses true',
 'good character',
```

```
 'good cook',
 'good dressing',
 'good memory',
 'govt emp',
 'gym',
 'h color black',
 'hair brown',
 'hair color black',
 'hair type silky',
 'hairstylist',
 'handsome',
 'handwriting',
 'happy',
 'hard working',
 'hardworking',
 'height cm',
 'height meter',
 'helpful',
 'helping',
 'helping nature',
 'hero',
 'highly competetive',
 'highly intelligent',
 'hobbies football comics',
 'holy person',
 'honest',
 'humble',
 'humorous',
 'humour',
 'hyper',
 'hypertrophy',
 'ignorance',
 'ignorant',
 'innovative',
 'inspiring',
 'intellegent',
 'intelligent',
 'intriguing',
 'introvert',
 'job asstt professor',
 'jolly',
 'joyful',
 'kind',
 'knowledge',
 'lame',
 'lazy',
 'legendary',
 'likes cricket',
 'likes movies',
```

```
 'listen song',
 'lovable',
 'loving',
 'low marks',
 'loyal',
 'makes fun of everyone',
 'manga comic',
 'married',
 'mature',
 'mental',
 'mischievious',
 'motivator',
 'movies',
 'movies songs',
 'multi billonaire',
 'multitasking',
 'native lang telgu',
 'natural',
 'naughty',
 'netflix',
 'newspaper',
 'no emotions',
 'non controversial',
 'non vegetarian',
 'occupation jila sanyojak',
 'orthodox',
 'overcomig failure',
 'overconfident',
 'overthinker',
 'painter',
 'patience',
 'physicque',
 'piano',
 'pirate',
 'polite',
 'politician',
 'pretty',
 'pretty eyes',
 'professional',
 'progressive mind',
 'puntual',
 'reader',
 'reading',
 'reasonable',
 'relational',
 'rich',
 'romantic',
 'sacrifice',
 'sarcastic',
```

```
'self confidance',
'selfless',
'serious',
'short',
'short hair',
'short height',
'short temper',
'short tempered',
'shortcut study',
'shorter',
'shy',
'simple',
'singer',
'singing',
'sister',
'sketcher',
'skill computer language',
'skill play with keyboard',
'skin color brown',
'skin type dry',
'skintone pale',
'sleepy',
'slim',
'small',
'small hair',
'smart',
'social',
'socialist',
'socially dead',
'space enthusiast',
'space exploration',
'sport',
'sports',
'sportsman',
'standup comedian',
'story writer',
'strength',
'strict',
'strong',
'support',
'supportive',
'sweet',
'table tennis',
'tall',
'tallest',
'teacher',
'teddy bear',
'tempered',
'thin',
```

```
 'thoughtful',
 'traveling',
 'trustworthy',
 'understanding',
 'very arogent',
 'very emotional',
 'video game',
 'vijaywada ap',
 'vizag',
 'voilent',
 'weak communication',
 'white skin',
 'working true',
 'workout',
 'writer',
 'writing',
 'years old']

input_data = " ".join(cleaned_data)

wordcloud = WordCloud(
    width=1000, height=500,
    background_color="black",  # Better contrast
    colormap="coolwarm",        # Improved color palette
    stopwords=stop_words, # Remove unimportant words
    contour_color='white',      # Adds an outline
    contour_width=1.5,
    max_words=100,              # Limit number of words
    max_font_size=100           # Improve font size
).generate(input_data)

# Display the word cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

Word Embedding

```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')
nltk.download('punkt_tab')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
```

```
True
```

```
import pandas as pd
import re
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import nltk
import numpy as np
import random
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

```
# Load the dataset
file_path = "/content/drive/MyDrive/Course Work/Sem 4/Data Analysis
and Visualization/Lab 6/Person_Data.xlsx"  # Update with correct path
df = pd.read_excel(file_path, sheet_name="Sheet1")

text_data = df.iloc[:, 2:].astype(str).values.flatten()

def preprocess_text(text, stop_words, extra_words):
  cleaned_text_data = [word for word in text_data if not
(isinstance(word, float) and np.isnan(word)) and word != "nan"]
  cleaned_words = set()
  for word in cleaned_text_data:
        word = word.lower().strip()  # Convert to lowercase and remove
extra spaces
        word = re.sub(r"[^a-z\s]", "", word)  # Remove numbers and
special characters
        word = re.sub(r"\s+", " ", word).strip()  # Remove extra
spaces

        if word and word not in stop_words and word not in
extra_words:
            cleaned_words.add(word)  # Add to set (removes duplicates)
  return sorted(list(cleaned_words))

stop_words = set(stopwords.words("english"))

extra_words = {
    "years", "old", "height", "meter", "kg", "cm", "color", "eye",
"hair",
    "male", "female", "working", "job", "govt", "assistant",
"professor",
    "true", "false", "children", "brown", "black", "white", "pale",
"dark",
    "student", "person", "occupation", "weight", "actor", "leo",
"english",
    "telugu", "japanese", "american", "native", "language", "study"
}

tokenized_text = preprocess_text(text_data, stop_words, extra_words)

tokenized_text
```

```
['age',
 'agrees to everything',
 'always panics',
 'ambitions',
 'angry',
 'animal lover',
 'animated show',
 'anime',
 'arrogant',
```

```
'artistic',
'atheletic figure',
'athletic',
'attractive',
'awareness',
'bad at sports',
'badminton',
'bargaining',
'basketball player',
'beautiful',
'best cook',
'best friend',
'beutiful',
'bhajan',
'big eyes',
'big face',
'bindi true',
'black eyes',
'bold',
'bulky body',
'business',
'calm',
'calm and composed',
'caring',
'casual in study',
'childish',
'chill',
'coder',
'color brown',
'communication',
'compromising',
'confident',
'conservative',
'consistent',
'cool',
'courageous',
'coward',
'creative',
'cricket',
'cry often',
'curly hair',
'cute',
'decision making',
'destructive',
'determined',
'disciplined',
'doing engineering',
'drive',
'driver',
```

```
'early riser',
'earring true',
'effortless',
'emotion less',
'emotionaally distant',
'emotional',
'englandactor',
'enjoyable',
'expressive',
'eye color black',
'eye color brown',
'eye dark brown',
'eye glasses',
'fair',
'famous',
'fast',
'fast thinking',
'fav color yellow',
'fav food upma',
'fav mov inception',
'fearless',
'feet',
'fit',
'flute',
'food biryani',
'foodie',
'friend',
'friendly',
'funny',
'geedy about new thing',
'gentle',
'girl',
'glasses true',
'good character',
'good cook',
'good dressing',
'good memory',
'govt emp',
'gym',
'h color black',
'hair brown',
'hair color black',
'hair type silky',
'hairstylist',
'handsome',
'handwriting',
'happy',
'hard working',
'hardworking',
```

```
 'height cm',
 'height meter',
 'helpful',
 'helping',
 'helping nature',
 'hero',
 'highly competetive',
 'highly intelligent',
 'hobbies football comics',
 'holy person',
 'honest',
 'humble',
 'humorous',
 'humour',
 'hyper',
 'hypertrophy',
 'ignorance',
 'ignorant',
 'innovative',
 'inspiring',
 'intellegent',
 'intelligent',
 'intriguing',
 'introvert',
 'job asstt professor',
 'jolly',
 'joyful',
 'kind',
 'knowledge',
 'lame',
 'lazy',
 'legendary',
 'likes cricket',
 'likes movies',
 'listen song',
 'lovable',
 'loving',
 'low marks',
 'loyal',
 'makes fun of everyone',
 'manga comic',
 'married',
 'mature',
 'mental',
 'mischievious',
 'motivator',
 'movies',
 'movies songs',
 'multi billonaire',
```

```
 'multitasking',
 'native lang telgu',
 'natural',
 'naughty',
 'netflix',
 'newspaper',
 'no emotions',
 'non controversial',
 'non vegetarian',
 'occupation jila sanyojak',
 'orthodox',
 'overcomig failure',
 'overconfident',
 'overthinker',
 'painter',
 'patience',
 'physicque',
 'piano',
 'pirate',
 'polite',
 'politician',
 'pretty',
 'pretty eyes',
 'professional',
 'progressive mind',
 'puntual',
 'reader',
 'reading',
 'reasonable',
 'relational',
 'rich',
 'romantic',
 'sacrifice',
 'sarcastic',
 'self confidance',
 'selfless',
 'serious',
 'short',
 'short hair',
 'short height',
 'short temper',
 'short tempered',
 'shortcut study',
 'shorter',
 'shy',
 'simple',
 'singer',
 'singing',
 'sister',
```

```
 'sketcher',
 'skill computer language',
 'skill play with keyboard',
 'skin color brown',
 'skin type dry',
 'skintone pale',
 'sleepy',
 'slim',
 'small',
 'small hair',
 'smart',
 'social',
 'socialist',
 'socially dead',
 'space enthusiast',
 'space exploration',
 'sport',
 'sports',
 'sportsman',
 'standup comedian',
 'story writer',
 'strength',
 'strict',
 'strong',
 'support',
 'supportive',
 'sweet',
 'table tennis',
 'tall',
 'tallest',
 'teacher',
 'teddy bear',
 'tempered',
 'thin',
 'thoughtful',
 'traveling',
 'trustwarthy',
 'understanding',
 'very arogent',
 'very emotional',
 'video game',
 'vijaywada ap',
 'vizag',
 'voilent',
 'weak communication',
 'white skin',
 'working true',
 'workout',
 'writer',
```

```
 'writing',
 'years old']

class KMeansTextClustering:
    def __init__(self, k=3, max_iters=100, tol=1e-4):
        self.k = k
        self.max_iters = max_iters
        self.tol = tol  # Tolerance for convergence

    def fit(self, words):
        # Convert words to numerical representation using TF-IDF
        self.vectorizer = TfidfVectorizer()
        word_vectors = self.vectorizer.fit_transform(words).toarray()

        # Randomly initialize centroids
        np.random.seed(42)
        self.centroids = word_vectors[np.random.choice(len(words),
self.k, replace=False)]

        for _ in range(self.max_iters):
            # Assign each word to the nearest centroid using cosine
similarity
            similarities = cosine_similarity(word_vectors,
self.centroids)
            self.labels_ = np.argmax(similarities, axis=1)

            # Compute new centroids
            new_centroids = np.array([word_vectors[self.labels_ ==
i].mean(axis=0) for i in range(self.k)])

            # Check for convergence
            if np.linalg.norm(new_centroids - self.centroids) <
self.tol:
                break

            self.centroids = new_centroids  # Update centroids

        self.words = words
        self.word_vectors = word_vectors

    def get_cluster_words(self):
        clusters = {i: [] for i in range(self.k)}
        for word, cluster in zip(self.words, self.labels_):
            clusters[cluster].append(word)
        return clusters

    def get_cluster_centroids(self):
        # Find the closest word to each centroid
        centroids_as_words = []
        for centroid in self.centroids:
```

```
            similarities = cosine_similarity([centroid],
self.word_vectors)[0]
            closest_word_index = np.argmax(similarities)
            centroids_as_words.append(self.words[closest_word_index])
        return centroids_as_words

# Run K-Means for Text Clustering
kmeans = KMeansTextClustering(k=3)

kmeans.fit(tokenized_text)

clusters = kmeans.get_cluster_words()

print("\nWord Clusters:")
for cluster_id, word_list in clusters.items():
    print(f"Cluster {cluster_id}: {word_list}")
```

```
Word Clusters:
Cluster 0: ['age', 'agrees to everything', 'always panics',
'ambitions', 'angry', 'animal lover', 'anime', 'arrogant', 'artistic',
'atheletic figure', 'athletic', 'attractive', 'awareness', 'bad at
sports', 'badminton', 'bargaining', 'basketball player', 'beautiful',
'best cook', 'best friend', 'beutiful', 'bhajan', 'big eyes', 'big
face', 'bindi true', 'black eyes', 'bold', 'bulky body', 'business',
'calm', 'calm and composed', 'caring', 'casual in study', 'childish',
'chill', 'coder', 'color brown', 'communication', 'compromising',
'confident', 'conservative', 'consistent', 'cool', 'courageous',
'coward', 'creative', 'cricket', 'cry often', 'curly hair', 'cute',
'decision making', 'destructive', 'determined', 'disciplined', 'doing
engineering', 'drive', 'driver', 'early riser', 'earring true',
'effortless', 'emotion less', 'emotionaally distant', 'emotional',
'englandactor', 'enjoyable', 'expressive', 'eye color black', 'eye
color brown', 'eye dark brown', 'eye glasses', 'fair', 'famous',
'fast', 'fast thinking', 'fav color yellow', 'fav food upma', 'fav mov
inception', 'fearless', 'fit', 'flute', 'food biryani', 'foodie',
'friend', 'friendly', 'funny', 'geedy about new thing', 'gentle',
'girl', 'glasses true', 'good character', 'good cook', 'good
dressing', 'good memory', 'govt emp', 'gym', 'h color black', 'hair
brown', 'hair color black', 'hair type silky', 'hairstylist',
'handsome', 'handwriting', 'happy', 'hard working', 'hardworking',
'height cm', 'height meter', 'helpful', 'helping', 'helping nature',
'hero', 'highly competetive', 'highly intelligent', 'hobbies football
comics', 'holy person', 'honest', 'humble', 'humorous', 'humour',
'hyper', 'hypertrophy', 'ignorance', 'ignorant', 'innovative',
'inspiring', 'intellegent', 'intelligent', 'intriguing', 'introvert',
'job asstt professor', 'jolly', 'joyful', 'kind', 'knowledge', 'lame',
'lazy', 'legendary', 'likes cricket', 'likes movies', 'listen song',
'lovable', 'loving', 'low marks', 'loyal', 'makes fun of everyone',
'manga comic', 'married', 'mature', 'mental', 'mischievious',
```

```
'motivator', 'movies', 'movies songs', 'multi billonaire',
'multitasking', 'native lang telgu', 'natural', 'naughty', 'netflix',
'newspaper', 'no emotions', 'non controversial', 'non vegetarian',
'occupation jila sanyojak', 'orthodox', 'overcomig failure',
'overconfident', 'overthinker', 'painter', 'patience', 'physicque',
'piano', 'pirate', 'polite', 'politician', 'pretty', 'pretty eyes',
'professional', 'progressive mind', 'puntual', 'reader', 'reading',
'reasonable', 'relational', 'rich', 'romantic', 'sacrifice',
'sarcastic', 'self confidance', 'selfless', 'serious', 'short', 'short
hair', 'short height', 'short temper', 'short tempered', 'shortcut
study', 'shorter', 'shy', 'simple', 'singer', 'singing', 'sister',
'sketcher', 'skill computer language', 'skill play with keyboard',
'skin color brown', 'skin type dry', 'skintone pale', 'sleepy',
'slim', 'small', 'small hair', 'smart', 'social', 'socialist',
'socially dead', 'space enthusiast', 'space exploration', 'sport',
'sports', 'sportsman', 'standup comedian', 'story writer', 'strength',
'strict', 'strong', 'support', 'supportive', 'sweet', 'table tennis',
'tall', 'tallest', 'teacher', 'teddy bear', 'tempered', 'thin',
'thoughtful', 'traveling', 'trustwarthy', 'understanding', 'very
arogent', 'very emotional', 'video game', 'vijaywada ap', 'vizag',
'voilent', 'weak communication', 'white skin', 'working true',
'workout', 'writer', 'writing', 'years old']
Cluster 1: ['animated show']
Cluster 2: ['feet']

centroids = kmeans.get_cluster_centroids()

print("\nCluster Centroids (Closest Representative Word):")
for cluster_id, centroid in enumerate(centroids):
    print(f"Cluster {cluster_id} Centroid: {centroid}")


Cluster Centroids (Closest Representative Word):
Cluster 0 Centroid: hair color black
Cluster 1 Centroid: animated show
Cluster 2 Centroid: feet
```