

```

import numpy as np
import pandas as pd

import matplotlib.pyplot as plt

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import sigmoid_kernel

# Read the CSV file
df = pd.read_csv("/content/cement.csv")

applications = df['Application'].unique()
print(applications)
# Take search material input from the user and convert it to lowercase
search_material = input("Enter the search Material: ")
search_material = search_material.lower()
search_material

['Foundation' 'Walls' 'Roofing' 'Flooring' 'Plastering' 'Repairs'
'Mass Concrete']
Enter the search Material: opc33
'opc33'

def filter_material(application, material):
    c_df = df[df['Application'] == application]
    neces_df = c_df[['Material', 'Review']] # Select Material and Review
    tfv = TfidfVectorizer(min_df=2, max_features=None,
                          strip_accents='unicode', analyzer='word',
                          token_pattern=r'\w{1,}', ngram_range=(1, 3),
                          stop_words='english')
    neces_df.loc[:, 'Review'] = neces_df['Review'].fillna('')
    tfv_matrix = tfv.fit_transform(neces_df['Review'])
    sig = sigmoid_kernel(tfv_matrix, tfv_matrix)
    indices = pd.Series(neces_df.index, index=neces_df['Material']).drop_duplicates()
    indices_lower = indices.index.str.lower()

    def give_rec(title, sig=sig):
        if title in neces_df['Material'].tolist():
            # Matched Title: Return similar materials for the exact title
            idx = indices[indices_lower.get_loc(title)]
            sig_scores = list(enumerate(sig[idx]))
            sig_scores = sorted(sig_scores, key=lambda x: x[1], reverse=True)
            sig_scores = sig_scores[1:-1] # Exclude the material itself
            n_in = [i[0] for i in sig_scores]
            return df.iloc[n_in] # Return data from original df using n_in
        else:
            # Unmatched Title: Return application-specific materials sorted by similarity
            all_materials = neces_df.copy()
            all_materials['Similarity Score'] = sig.diagonal()
            all_materials = all_materials.sort_values(by='Similarity Score', ascending=False)
            all_materials.drop('Similarity Score', axis=1, inplace=True)
            return all_materials.merge(df[df['Application'] == application], how='left', on='Material') # Merge with filtered df for the specified

    # Call give_rec within filter_material (remove printing)
    all_materials = give_rec(material)

    return all_materials # Return the DataFrame containing recommendations

# List of applications (replace with your actual list)

# Loop through applications and call filter_material
for application in applications:
    if df[df['Application'] == application].shape[0] < 3:
        continue
    all_foundations = filter_material(application, search_material)

    # Print the DataFrame to inspect its structure
    print(f""" Recommendations for {application.upper()} Application:"")
    print(all_foundations)

    # Check if 'Similarity Score' column exists in all_foundations DataFrame
    if 'Similarity Score' in all_foundations.columns:
        plt.figure(figsize=(10, 6))
        plt.bar(all_foundations['Material'], all_foundations['Similarity Score'])

```

```

plt.xlabel('Material')
plt.ylabel('Similarity Score')
plt.title(f'Recommendations for {application.upper()} Application')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
else:
    print(f"No 'Similarity Score' column found in recommendations for {application.upper()} Application")

```

**\*\* Recommendations for FOUNDATION Application:**

	Material	Review_x \
0	PPC	"Offers enhanced durability for foundations."
1	Rapid Hardening	"Ideal for quick-setting foundations."
2	Extra Rapid Hardening	"Provides exceptionally fast setting for found..."
3	Quick Setting	"Sets rapidly for speedy foundations."
4	Low Heat	"Generates less heat during hydration, ideal ..."
5	Sulphate Resisting	"Resists sulphate attacks, ideal for foundati..."
6	OPC43	"Suitable for high-strength foundations, ensu..."
7	OPC53	"Ideal for high-strength foundations, ensures..."
8	OPC33	"Ideal for general-purpose foundation works."

	Application	Strength	Durability	Availability	Cost (per bag) \
0	Foundation	High	High	High	340
1	Foundation	High	High	High	360
2	Foundation	High	High	High	380
3	Foundation	High	High	High	320
4	Foundation	High	High	High	400
5	Foundation	High	High	High	420
6	Foundation	High	High	High	380
7	Foundation	High	High	High	400
8	Foundation	Medium	Medium	High	340

	Environmental	Impact	Compatibility	Ease of Use	Brand/Supplier \
0	Medium	Compatible	Easy	ACC Cement	
1	Medium	Compatible	Moderate	ACC Cement	
2	Medium	Compatible	Moderate	ACC Cement	
3	Medium	Compatible	Moderate	ACC Cement	
4	Medium	Compatible	Moderate	ACC Cement	
5	Medium	Compatible	Moderate	ACC Cement	
6	Medium	Compatible	Moderate	ACC Cement	
7	Medium	Compatible	Moderate	ACC Cement	
8	Medium	Compatible	Moderate	ACC Cement	

	Review_y	Ratings
0	"Offers enhanced durability for foundations."	4.2
1	"Ideal for quick-setting foundations."	4.2
2	"Provides exceptionally fast setting for found..."	4.0
3	"Sets rapidly for speedy foundations."	4.4
4	"Generates less heat during hydration, ideal ..."	4.3
5	"Resists sulphate attacks, ideal for foundati..."	4.2
6	"Suitable for high-strength foundations, ensu..."	4.2
7	"Ideal for high-strength foundations, ensures..."	4.0
8	"Ideal for general-purpose foundation works."	4.5

No 'Similarity Score' column found in recommendations for FOUNDATION Application

**\*\* Recommendations for WALLS Application:**

	Material	Review_x \
0	PPC	"Suitable for strong and resilient walls."
1	Rapid Hardening	"Suitable for rapid construction of walls."
2	Extra Rapid Hardening	"Ideal for quick construction of walls."
3	Quick Setting	"Ideal for quick construction of walls."
4	Low Heat	"Suitable for structures where heat build-up i..."
5	Sulphate Resisting	"Provides excellent protection against sulfate..."
6	OPC43	"Provides strong and resilient walls, perfect..."
7	OPC53	"Provides superior strength for resilient wall..."
8	OPC33	"Suitable for standard wall constructions."

	Application	Strength	Durability	Availability	Cost (per bag) \
--	-------------	----------	------------	--------------	------------------

Start coding or [generate](#) with AI.

