DataPlay Reviews Word Cloud using NLP

Author

DINESH S

my linkedin profile | github link | Data Play

Task 1:

```
1.DataPlay Reviews WordCloud
Level 1:"Using Excel Functions/Pivot Table getting word
frequency count, removing stopwords"
Level 2:Use Tf-Idf
Level 3:Now implement (Dictionary of words with their
frequency of occurrence) word cloud in Power BI
```

Overview

This notebook demonstrates how to process and analyze review data from DataPlay using Python. The main objectives are to:

- **Load and preprocess the data**: Combine data from a CSV file (with words in separate cells), clean the text, and remove stopwords.
- **Compute word frequencies and/or TF-IDF scores**: Generate a dictionary (or table) of words along with their frequency counts or TF-IDF weights.
- **Visualize the results with a Word Cloud**: Use the `wordcloud` and `matplotlib` libraries to create a visual representation of the most significant words in the reviews.

Prerequisites

Before running the notebook, ensure you have the following Python packages installed:

- **pandas**: For data manipulation and CSV file reading.
- **nltk**: For natural language processing tasks such as stopwords removal.
- **wordcloud**: To generate the word cloud visualization.
- **matplotlib**: To display the generated word cloud.
- **re**: For regular expression operations.

You can install the required packages using pip:

```
pip install pandas nltk wordcloud matplotlib
```

Additionally, the notebook downloads necessary NLTK data (stopwords) if not already present.

In [ ]:

```
#installing required Dependencies
!pip install pandas nltk wordcloud matplotlib
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-pack
ages (2.2.2)
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packag
es (3.9.1)
Requirement already satisfied: wordcloud in /usr/local/lib/python3.11/dist-p
ackages (1.9.4)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-
packages (3.10.0)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/di
st-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/pyth
on3.11/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dis
t-packages (from pandas) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/d
ist-packages (from pandas) (2025.1)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packa
ges (from nltk) (8.1.8)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-pack
ages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.11/
dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packag
es (from nltk) (4.67.1)
Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-pack
ages (from wordcloud) (11.1.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.1
1/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dis
t-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.1
1/dist-packages (from matplotlib) (4.55.8)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.1
1/dist-packages (from matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/
dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.1
1/dist-packages (from matplotlib) (3.2.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-pa
ckages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```

In [ ]:

```python
#initializing Dependencies
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

In [ ]:

```python
#initializing Dependencies for NLP
import re
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from wordcloud import WordCloud
from collections import Counter
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [ ]:

```
# Download required NLTK resources
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('punkt_tab')
```

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

Out[ ]:

```
True
```

Step 1: Load the CSV File

In [ ]:

```
df = pd.read_csv('/content/DataPlay_Reviews_unique_keyword - unique_keyword.
```

In [ ]:

```
df.head(5)
```

Out[ ]:

	0	1	2	3	4	5	6	7
0	had	a	fantastic	experience	at	DataPlay.	The	institute
1	to	me	it's	a	very	good	place	for
2	sir	and	Mahima	ma'am	have	outstanding	sessions	that
3	it	has	been	a	great	experience,	the	mentors
4	days	agoNew\nMy	overall	experience	was	great.	Mentors	were

5 rows × 77 columns

Step 2: Data Cleaning and Preprocessing

Text preprocessing

In [ ]:

```
# Combine all words from the DataFrame into a single text string
all_words = " ".join(df.fillna('').astype(str).values.flatten())
```

In [ ]:

```
# Convert to lowercase and remove punctuation
clean_text = re.sub(r'[^\w\s]', '', all_words.lower())
```

Remove stopwords

In [ ]:

```
# Remove stopwords
stop_words = set(stopwords.words('english'))
filtered_words = [word for word in clean_text.split() if word not in stop_wo
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [ ]:

```
stop_words
```

```
Out[ ]: {'a',
         'about',
         'above',
         'after',
         'again',
         'against',
         'ain',
         'all',
         'am',
         'an',
         'and',
         'any',
         'are',
         'aren',
         "aren't",
         'as',
         'at',
         'be',
         'because',
         'been',
         'before',
         'being',
         'below',
         'between',
         'both',
         'but',
         'by',
         'can',
         'couldn',
         "couldn't",
         'd',
         'did',
         'didn',
         "didn't",
         'do',
         'does',
         'doesn',
         "doesn't",
         'doing',
         'don',
         "don't",
         'down',
         'during',
         'each',
         'few',
         'for',
         'from',
         'further',
         'had',
         'hadn',
         "hadn't",
         'has',
         'hasn',
         "hasn't",
         'have',
            "haven't",
            'having',
            'he',
            'her',
            'here',
            'hers',
            'herself',
            'him',
            'himself',
            'his',
            'how',
            'i',
            'if',
            'in',
            'into',
            'is',
            'isn',
            "isn't",
            'it',
            "it's",
            'its',
            'itself',
            'just',
            'll',
            'm',
            'ma',
            'me',
            'mightn',
            "mightn't",
            'more',
            'most',
            'mustn',
            "mustn't",
            'my',
            'myself',
            'needn',
            "needn't",
            'no',
            'nor',
            'not',
            'now',
            'o',
            'of',
            'off',
            'on',
            'once',
            'only',
            'or',
            'other',
            'our',
            'ours',
            'ourselves',
            'out',
            'over',
            'own',
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
 's',
 'same',
 'shan',
 "shan't",
 'she',
 "she's",
 'should',
 "should've",
 'shouldn',
 "shouldn't",
 'so',
 'some',
 'such',
 't',
 'than',
 'that',
 "that'll",
 'the',
 'their',
 'theirs',
 'them',
 'themselves',
 'then',
 'there',
 'these',
 'they',
 'this',
 'those',
 'through',
 'to',
 'too',
 'under',
 'until',
 'up',
 've',
 'very',
 'was',
 'wasn',
 "wasn't",
 'we',
 'were',
 'weren',
 "weren't",
 'what',
 'when',
 'where',
 'which',
 'while',
 'who',
 'whom',
 'why',
 'will',
 'with',
 'won',
 "won't",
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
 "wouldn't",
 'y',
 'you',
 "you'd",
 "you'll",
 "you're",
 "you've",
 'your',
 'yours',
 'yourself',
 'yourselves'}
```

In [ ]:

```python
# Join the filtered words back into a single string
processed_text = " ".join(filtered_words)
processed_text
```

```
Out[ ]: 'fantastic experience dataplay institute offers excellent training data ana
        lysis covering statistics excel operations power bi tools knowledgeable ins
        tructors comprehensive materials make top choice aspiring data scientists h
        ighly recommended quality education handson learning good place learning go
        od hearted teachers institutes put efforts towards students average datapla
        y put efforts every single student teaching style nice im new course didnt 
        wonder softly understand every single thing thought possible student friend
        ly teaching sir mahima maam outstanding sessions help gain clarity improve 
        skills great experience mentors really helpful well job making classes enga
        ging interactive days agonew overall experience great mentors incredibly su
        pportive effectively explaining tools concepts reallife scenariosi gained v
        aluable handson experience power bi excel engaged discussions interview que
        stions significantly improved knowledge throughout course explanations star
        t basics content easy understand assignments provide route application conc
        epts excellent learning place aspiring data scientist data analyst currentl
        y enrolled data science training programthe mentors truly good hearted expe
        rienced professionals provide valuable guidance helps every student small p
        roblemthe handson learning approach supportive environment make top choice 
        entering field data science data analyst overall experience great currently
        learning data analysis going well started basics good pace sessions interac
        tive good place start recommended nishant sir mahima maam highly motivating
        insightful working nishant sir mahima maam incredible guidance support help
        ed improvise achieve personal professional goals session amazing insightful
        empowering nishant sir mahima maam outstanding sessions help gain clarity i
        mprove skills personalized sessions constructive feedback improve every ste
        p join dataplay nice experience beginners easy understand nice experience g
        reat features good experience easy explain currently enrolled data analysis
        data science training program dataplay couldnt thrilled experience mentors 
        truly exceptional consistently going beyond ensure every student thoroughly
        understands concept experience really good sir mam helpful gained much valu
        able experience mentors dataplay exceptionally productive focused multidime
        nsional learning ensure stay updated latest industry trends developments co
        mmitment understanding students strengths weaknesses remarkable tailor teac
        hing methods cultivate deeper understanding concepts helping us grasp compl
        ex topics thoroughly confidently data analytics training transformative exp
        erience instructors incredibly knowledgeable engaging making complex concep
        ts easy understand handson exercises realworld case studies provided invalu
        able practical skills plus supportive learning environment fostered collabo
        ration growth highly recommend training anyone looking excel field data ana
        lytics deeply grateful exceptional utility dataplay provides remarkably use
        rfriendly efficient platform analytical tasks moreover teaching atmosphere 
        characterized friendly environment making learning experience truly enjoyab
        le gem data science world sure personalized approach realworld insights mak
        e learning feel natural engaging daily practice problems gamechanger deepen
        ing understanding whether youre beginner pro real deal leveling skills high
        ly recommend offers topnotch education effective teaching methods practical
        exercises realworld examples enabling students gain confidence excel data s
        cience analysis great initiative nishant mahima teach folks outstanding tea
        ching skills offline leactures good understanding ask doubts meantors helpf
        ul experience learning mentor helpful good provided immersive learning expe
        rience wellstructured curriculum teachers good supportive wish extend since
        re gratitude exceptional utility dataplay find functionality remarkably use
        rfriendly efficient analytical tasks institutions place learning immersive 
        journey realm data science step institution youre greeted bustling atmosphe
        re charged intellectual curiosity innovation fundamentals statistics probab
                                  arning algorithms every aspect field covered preci
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
sion clarity instructors experts respective domains guide maze knowledge pa
tience enthusiasm place learn coding enhance data science great mentorship
guidance play excellent teaching institute aspiring data analysts mentors e
xperienced professionals provide valuable guidance study material comprehen
sive handson learning approach supportive environment make top choice enter
ing field data analysis diving data science tons resources support help mas
ter field chill environment also offer manageable timings making perfect wo
rking professionals plus regularly test knowledge keep track good organizat
ion work grow course good educational journey dataplay showcases promise dy
namic curriculum dedicated faculty shaping vibrant learning community good
organisation work grow nishant sir mam really helpful guiding work environm
ent good proper learning provided people nice place work people place work
learning place begin learning data science initiative mahima nishant really
enjoyed seminar organized appreciate effort putting train students data ana
latics data science track seminar'
```

Pivot Table getting word frequency count

In [ ]:

```python
# Create a DataFrame from word_frequencies
word_freq_df = pd.DataFrame.from_dict(word_frequencies, orient='index', colu
word_freq_df.index.name = 'Word'
word_freq_df = word_freq_df.reset_index()

# Create the pivot table
pivot_table = pd.pivot_table(word_freq_df, values='Frequency', index='Word',

# Display the pivot table
pivot_table
```

Out[ ]:

	Frequency
Word	
achieve	1
advanced	1
agonew	1
algorithms	1
also	1
...	...
wonder	1
work	5
working	2
world	1
youre	2

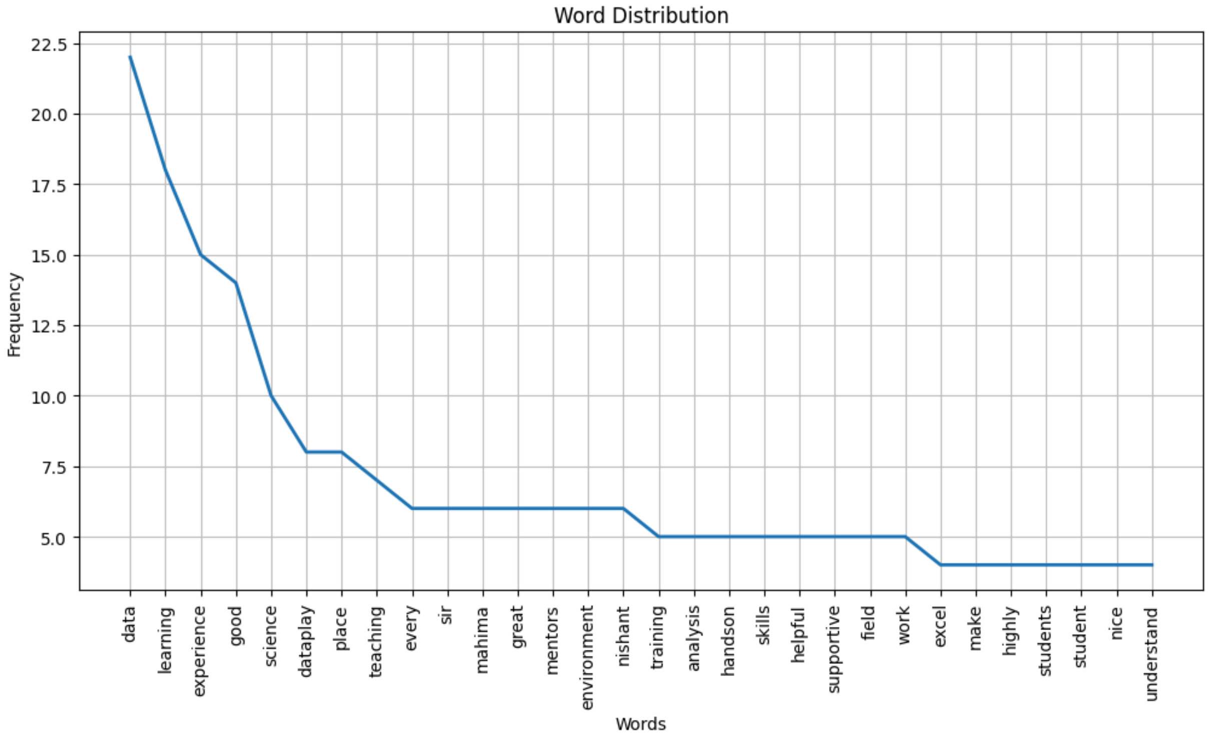
330 rows × 1 columns

Step 3: Data Analysis

In [ ]:

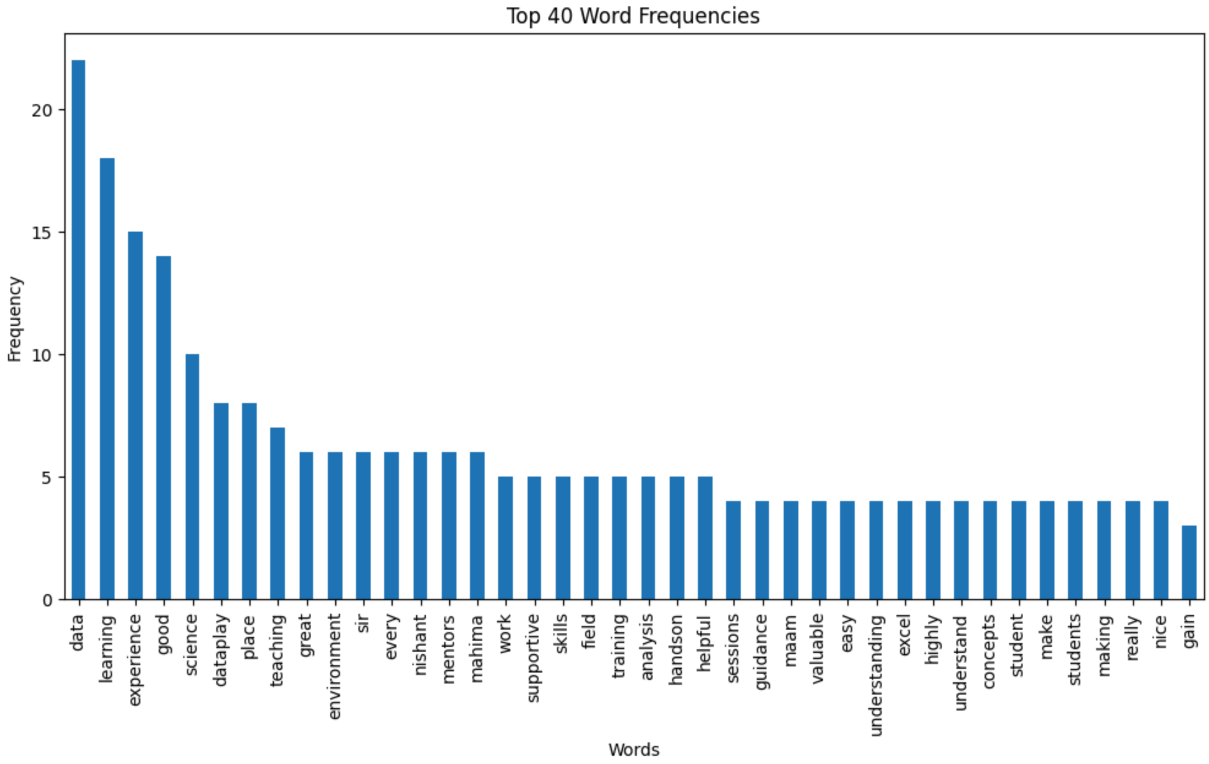
```
# Word Distribution
word_frequencies = nltk.FreqDist(filtered_words)

# Plot the word distribution
plt.figure(figsize=(12, 6))
word_frequencies.plot(30, cumulative=False)
plt.title('Word Distribution')
plt.xlabel('Words')
plt.ylabel('Frequency')
plt.show()
```



In [ ]:

```
# Histogram of word frequencies
word_counts = pd.Series(processed_text.split()).value_counts()
plt.figure(figsize=(12, 6))
word_counts[:40].plot(kind='bar')
plt.title('Top 40 Word Frequencies')
plt.xlabel('Words')
plt.ylabel('Frequency')
plt.show()
```



TF-IDF scores

In [ ]:

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform([processed_text]) # Fit and transfor

feature_names = vectorizer.get_feature_names_out()
tfidf_scores = tfidf_matrix.toarray()[0]
```

In [ ]:

```
# Create a DataFrame for TF-IDF scores
tfidf_df = pd.DataFrame({'Word': feature_names, 'TF-IDF Score': tfidf_scores

# Sort by TF-IDF score in descending order
tfidf_df = tfidf_df.sort_values(by='TF-IDF Score', ascending=False)

# Display the top N words with their TF-IDF scores
tfidf_df
```

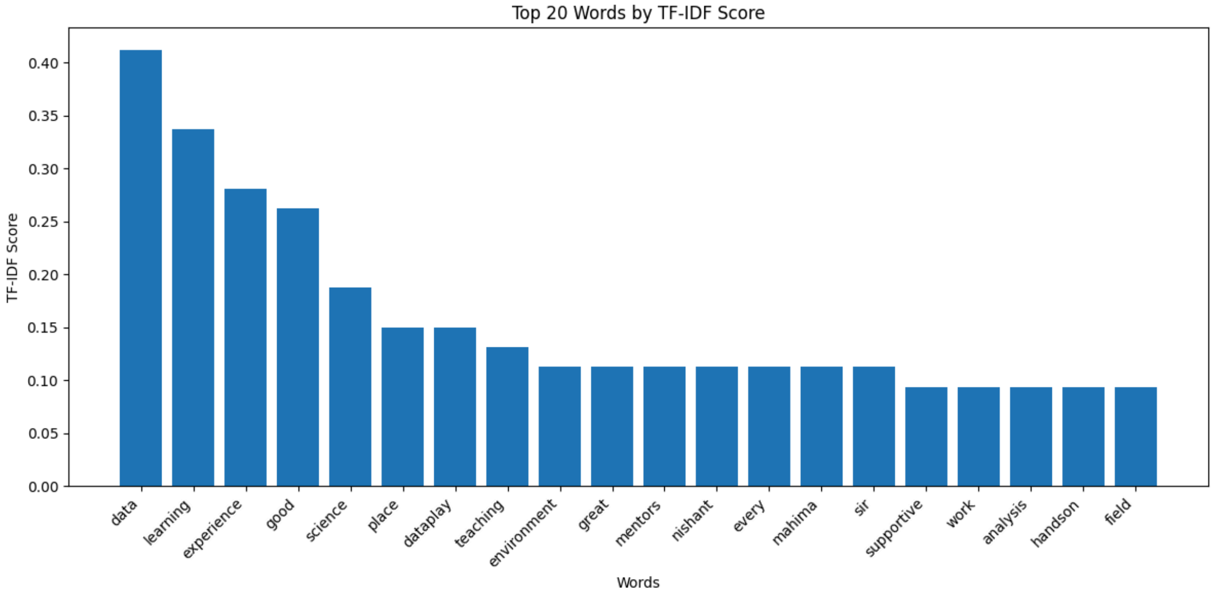
Out[ ]:

	Word	TF-IDF Score
58	data	0.412098
175	learning	0.337171
100	experience	0.280976
126	good	0.262244
258	science	0.187317
...	...	...
136	guiding	0.018732
140	helped	0.018732
142	helping	0.018732
143	helps	0.018732
165	invaluable	0.018732

330 rows × 2 columns

In [ ]:

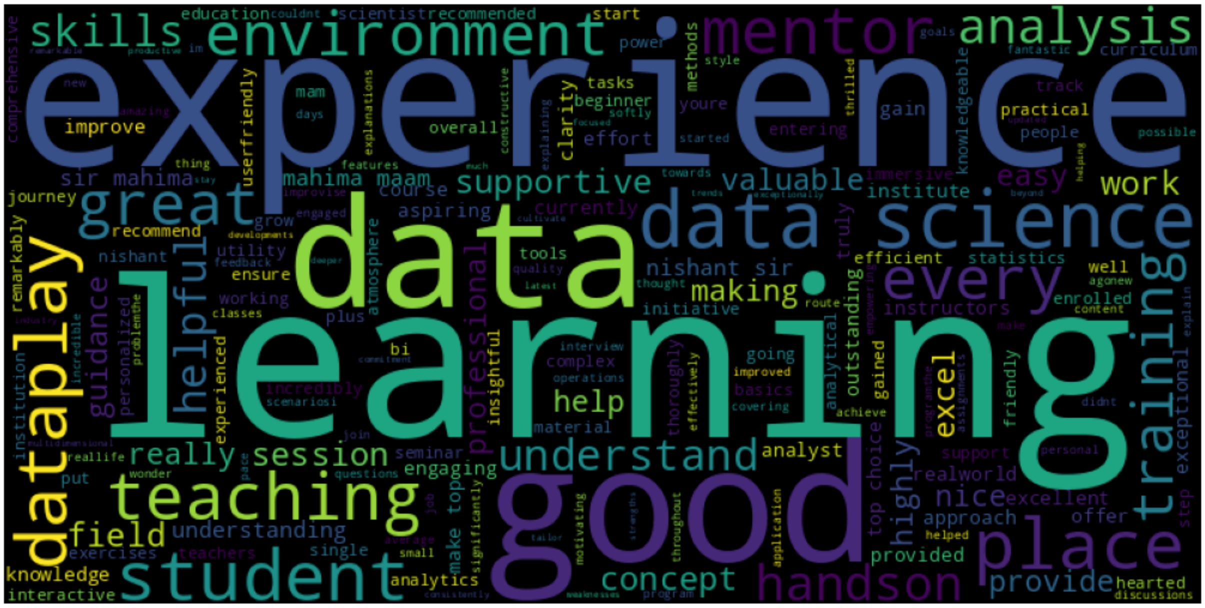
```
#create a bar plot of the top words
plt.figure(figsize=(12, 6))
plt.bar(tfidf_df['Word'][:20], tfidf_df['TF-IDF Score'][:20])
plt.xlabel('Words')
plt.ylabel('TF-IDF Score')
plt.title('Top 20 Words by TF-IDF Score')
plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels for better reada
plt.tight_layout()
plt.show()
```



Creating word cloud

```
In [ ]: # Generate the word cloud
        wordcloud = WordCloud(width=800, height=400, background_color='black').gener

        # Display the generated image:
        plt.figure(figsize=(10, 5), facecolor=None)
        plt.imshow(wordcloud)
        plt.axis("off")
        plt.tight_layout(pad=0)
        plt.show()
```



```
In [ ]: # Save the wordcloud image
        wordcloud.to_file("review_wordcloud.png")

Out[ ]: <wordcloud.wordcloud.WordCloud at 0x781eb0ca3150>
```

```
In [ ]: # save all the data

        # Save the pivot table to a CSV file
        pivot_table.to_csv('pivot_table.csv')

        # Save word frequencies to a CSV file
        word_freq_df.to_csv('word_frequencies.csv', index=False)

        # Save the TF-IDF DataFrame to a CSV file
        tfidf_df.to_csv('tfidf_scores.csv', index=False)
```

Conclusion

This notebook processes raw review data to create a meaningful word cloud. It demonstrates:

- Loading CSV data and handling data spread across multiple cells.
- Text cleaning, tokenization, and stopwords removal.
- Word frequency calculation (with an option to compute TF-IDF scores).

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- Visualization of the processed data with a word cloud.
- How to integrate your Python workflow into Power BI via Python visuals.

This documentation should help you (or anyone else reviewing the notebook) understand the purpose, methods, and steps involved in the analysis. It also provides guidance on leveraging Python within Power BI, ensuring that your work meets submission requirements without needing to rebuild the process entirely in Power BI’s native tools.

Acknowledgements

Special Thanks:
I would like to extend my heartfelt gratitude to DataPlay Company for the fellowship. This opportunity has been instrumental in enhancing my skills and enabling projects like this to flourish.

DATAPLAY

End of Notebook