MEOW

PROJECT TITLE - *KATY PERRY CSV*

TEAM NAM - *MEOW TOLI*

TEAM MEMBERS

1) AYUSH KUMAR MANDAL
2) ANIRUDH PANIGRAHI
3) VEDANSHA SRIVASTAVA
4) SHEKHAR NARAYAN MISHRA

INDIVIDUAL CONTRIBUTIONS

AYUSH KUMAR

Verified Zipf's Law and developed the supporting graphs.

```
from google.colab import files
uploaded = files.upload()
import pandas as pd

df = pd.read_csv('KatyPerry.csv')
df.head()
```

Choose files KatyPerry.csv

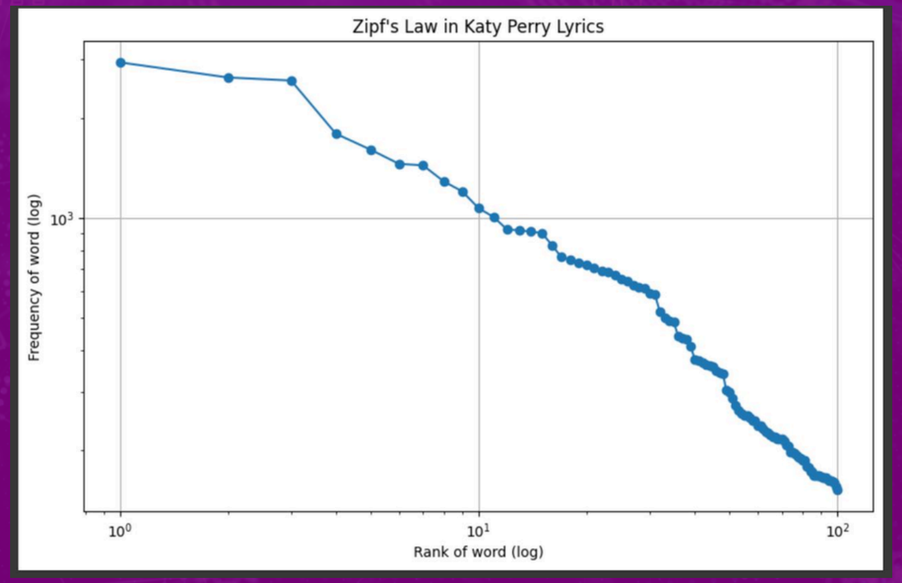
- **KatyPerry.csv**(text/csv) - 451321 bytes, last modified: 04/05/2025 - 100% done

Saving KatyPerry.csv to KatyPerry (2).csv

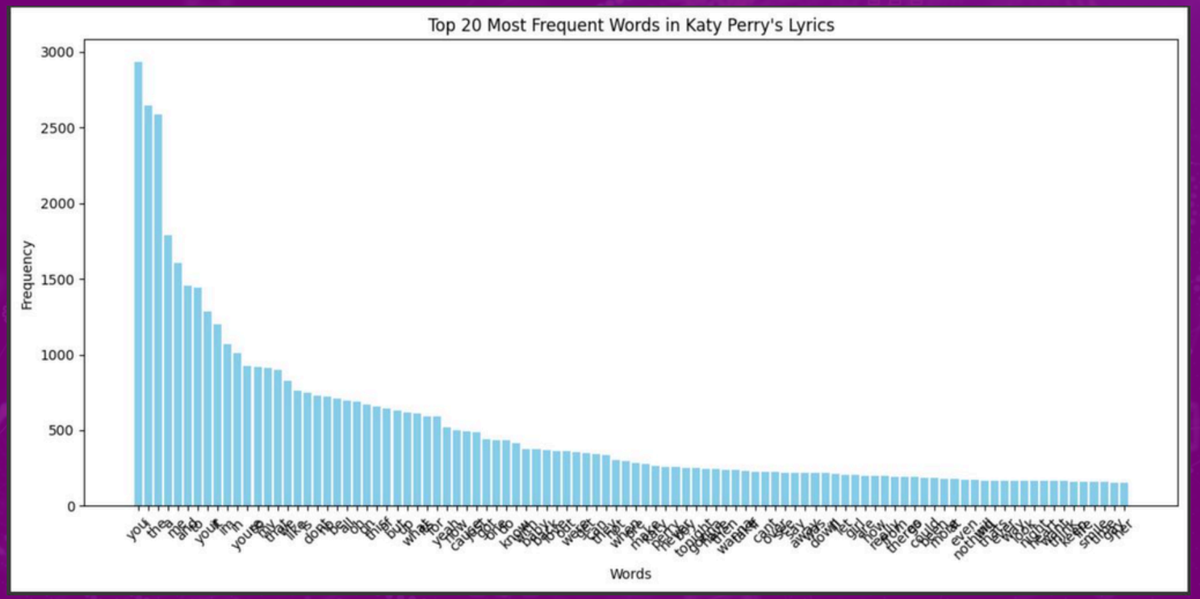
	Unnamed: 0	Artist	Title	Album	Year	Date	Lyric
0	0	Katy Perry	Swish Swish	Witness	2017.0	2017-05-19	refrain they know what is what but they don't ...
1	1	Katy Perry	Chained to the Rhythm	Witness	2017.0	2017-02-10	katy perry are we crazy living our lives throu...
2	2	Katy Perry	Dark Horse	PRISM	2013.0	2013-10-18	juicy j yeah ya'll know what it is katy perry ...
3	3	Katy Perry	Bon Appétit	Witness	2017.0	2017-04-28	quavo ayy yeah katy perry migos ayy katy per...
4	4	Katy Perry	Roar	PRISM	2013.0	2013-08-10	i used to bite my tongue and hold my breath sc...

```
[24] import re

    # Use the correct column name
    text = ' '.join(df['Lyric'].dropna().astype(str)).lower()

    # Remove punctuation and numbers
    text = re.sub(r'[^a-z\s]', '', text)

    # Tokenize (split into words)
    words = text.split()

    # Preview first 20 words
    print(words[:20])
```

```
from collections import Counter
import matplotlib.pyplot as plt
import numpy as np

# Get word frequencies
word_freq = Counter(words)
most_common = word_freq.most_common(100)

# Prepare data for Zipf plot
ranks = range(1, len(most_common) + 1)
frequencies = [freq for word, freq in most_common]

# Plot
plt.figure(figsize=(10, 6))
plt.plot(ranks, frequencies, marker='o')
plt.xscale('log')
plt.yscale('log')
plt.title("Zipf's Law in Katy Perry Lyrics")
plt.xlabel("Rank of word (log)")
plt.ylabel("Frequency of word (log)")
plt.grid(True)
plt.show()
```



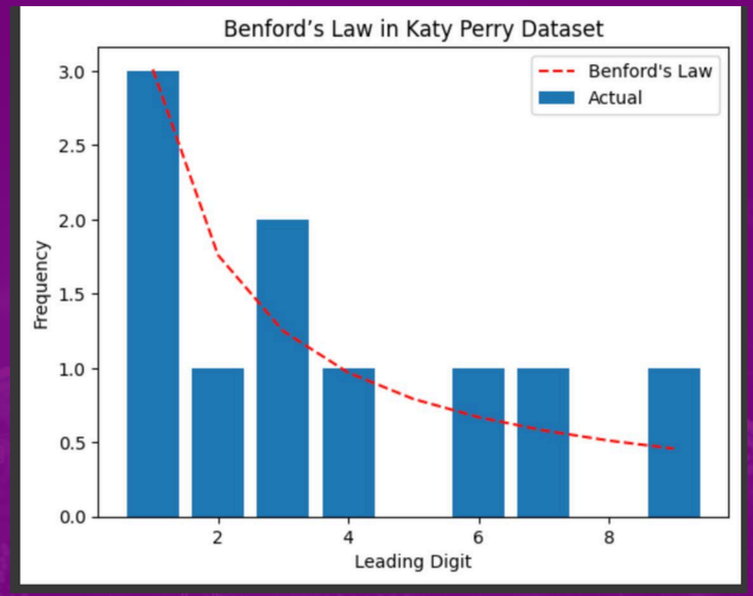
```python
# Prepare data
words, counts = zip(*most_common)

# Bar plot
plt.figure(figsize=(12, 6))
plt.bar(words, counts, color='skyblue')
plt.xticks(rotation=45)
plt.title("Top 20 Most Frequent Words in Katy Perry's Lyrics")
plt.xlabel("Words")
plt.ylabel("Frequency")
plt.tight_layout()
plt.show()
```



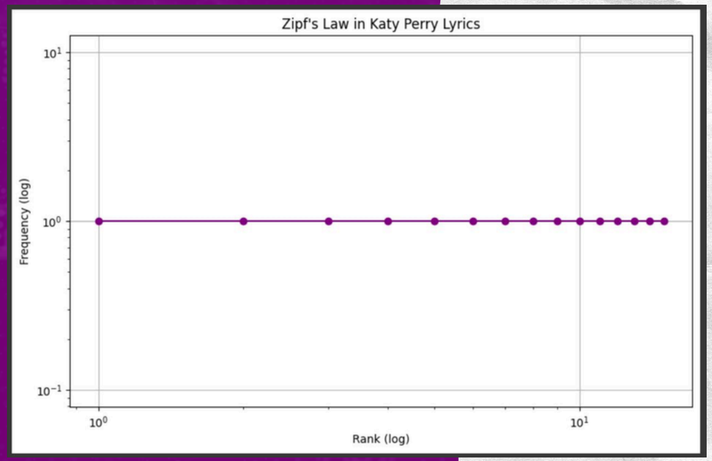
ANIRUDH PANIGRAHI

Investigated Brentford's Law, helped in Bonus Analysis and contributed to the PPT content.

```python
import matplotlib.pyplot as plt
import numpy as np

# Example dataset (replace with your actual data)
data = [123, 456, 789, 101, 202, 345, 678, 910, 111, 345]

# Function to extract the first digits
def extract_first_digit(numbers):
    return [int(str(n)[0]) for n in numbers if n > 0]

# Extract first digits
first_digits = extract_first_digit(data)

# Count actual frequencies
actual_counts = [first_digits.count(d) for d in range(1, 10)]

# Expected Benford's distribution
benford = [np.log10(1 + 1/d) for d in range(1, 10)]
expected_counts = [p * len(first_digits) for p in benford]

# Plot
plt.bar(range(1, 10), actual_counts, label="Actual")
plt.plot(range(1, 10), expected_counts, 'r--', label="Benford's Law")
plt.xlabel("Leading Digit")
plt.ylabel("Frequency")
plt.title("Benford's Law in Katy Perry Dataset")
plt.legend()
plt.show()
```




```python
#Zipf's Law Check (Log-Log Plot)
ranks = range(1, len(common_words)+1)
frequencies = [freq for word, freq in common_words]

plt.figure(figsize=(10, 6))
plt.plot(ranks, frequencies, 'o-', color='purple')
plt.xscale('log')
plt.yscale('log')
plt.title("Zipf's Law in Katy Perry Lyrics")
plt.xlabel("Rank (log)")
plt.ylabel("Frequency (log)")
plt.grid(True)
plt.show()
```

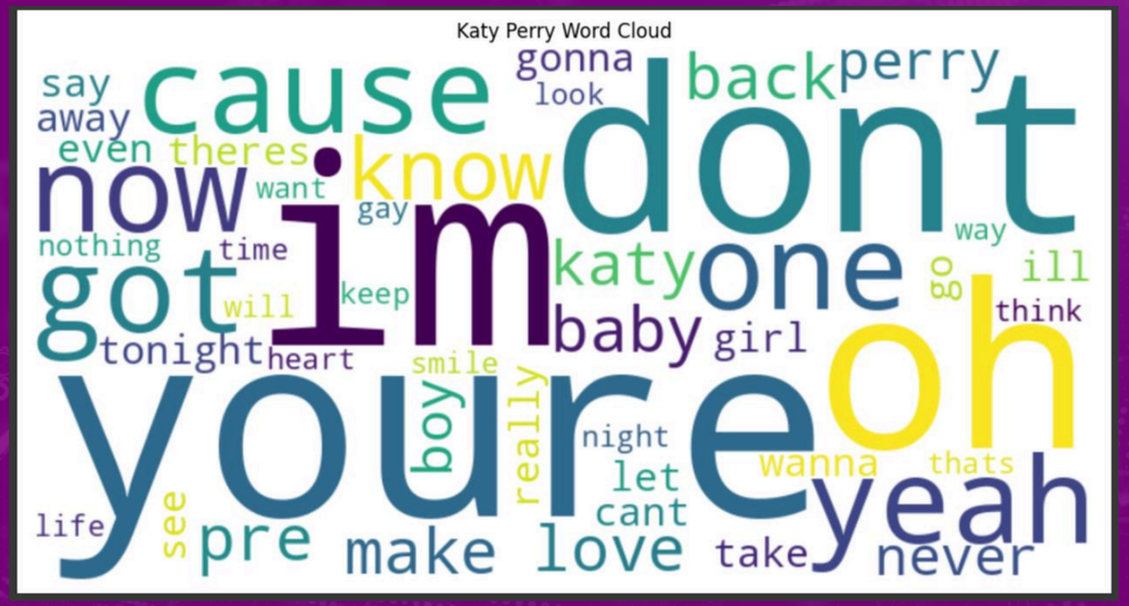


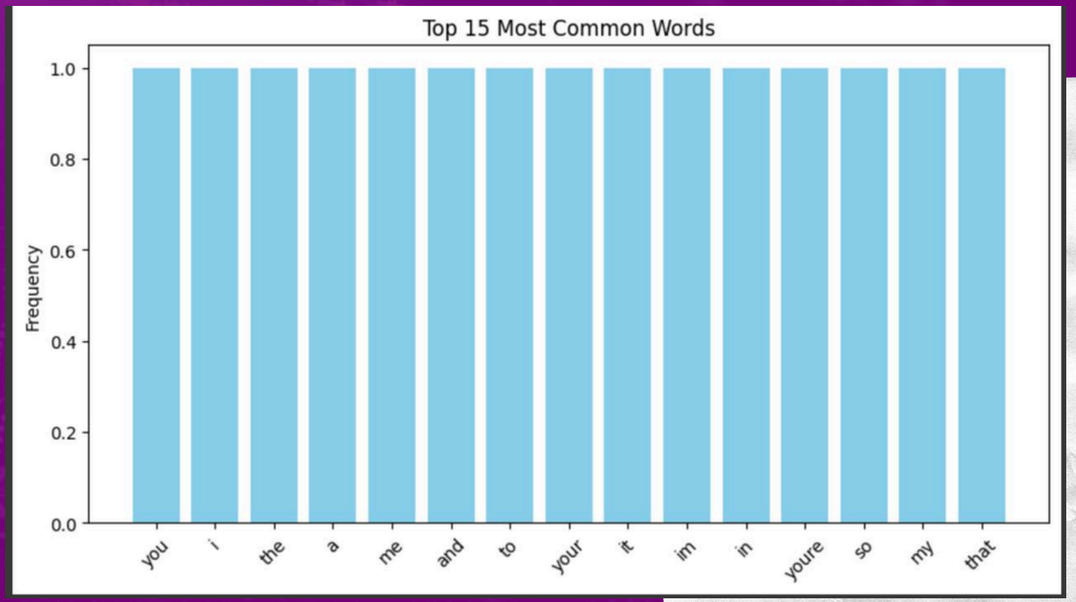
VEDANHSA SRIVASTAVA

Helped in building tools for analyzing bonus-related patterns.

```
#Word Cloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(' '.join(words))

plt.figure(figsize=(12, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title("Katy Perry Word Cloud")
plt.show()
```

```
#Top Words Bar Chart
word_freq = Counter(words)
common_words = word_freq.most_common(15)

words_list, counts = zip(*common_words)

plt.figure(figsize=(10, 5))
plt.bar(words_list, counts, color='skyblue')
plt.title("Top 15 Most Common Words")
plt.xticks(rotation=45)
plt.ylabel("Frequency")
plt.show()
```

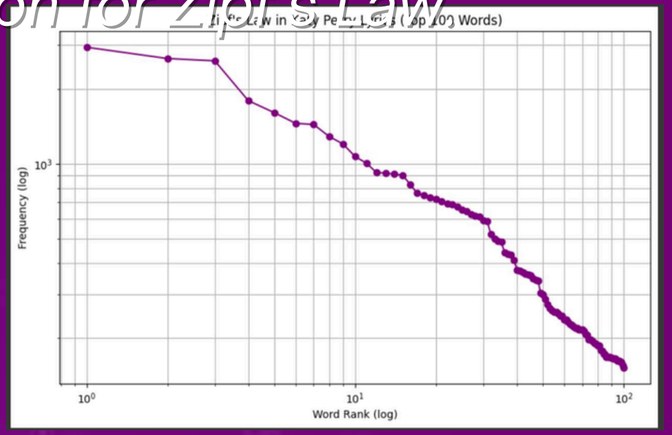




SHEKHAR NARAYAN MISHRA

Designed the PPT and its content, and created the line graph for Zipf's Law

```python
# Zipf's Law line graph (you ran or were guided to run this)
most_common_zipf = word_freq.most_common(100)
ranks = range(1, len(most_common_zipf) + 1)
frequencies = [count for word, count in most_common_zipf]

plt.figure(figsize=(10, 6))
plt.plot(ranks, frequencies, marker='o', linestyle='-', color='purple')
plt.xscale('log')
plt.yscale('log')
plt.title("Zipf's Law in Katy Perry Lyrics (Top 100 Words)")
plt.xlabel("Word Rank (log)")
plt.ylabel("Frequency (log)")
plt.grid(True, which='both')
plt.show()
```




```python
import re

# Use the correct column name
text = ' '.join(df['Lyric'].dropna().astype(str)).lower()

# Remove punctuation and numbers
text = re.sub(r'[^a-z\s]', '', text)

# Tokenize (split into words)
words = text.split()

# Preview first 20 words
print(words[:20])
```

```
['refrain', 'they', 'know', 'what', 'is', 'what', 'but', 'they', 'dont', 'know', 'what', 'is', 'what', 'they', 'just', 'strut', 'what', 'the', 'fuck', 'katy']
```

THANKYOU