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

Choose Files flipkart.csv

- **flipkart.csv**(text/csv) - 644271 bytes, last modified: 8/12/2024 - 100% done

```
Saving flipkart.csv to flipkart.csv
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
from wordcloud import WordCloud
```

```
!pip install vaderSentiment
```

```
Collecting vaderSentiment
  Downloading vaderSentiment-3.3.2-py2.py3-none-any.whl.metadata (572 bytes)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from vaderSentiment) (2.32.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->vaderSentiment) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->vaderSentiment) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->vaderSentiment) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->vaderSentiment) (2024.7.4)
Downloading vaderSentiment-3.3.2-py2.py3-none-any.whl (125 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 126.0/126.0 kB 2.4 MB/s eta 0:00:00
Installing collected packages: vaderSentiment
Successfully installed vaderSentiment-3.3.2
```

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import warnings
warnings.filterwarnings('ignore')
```

```
# Load the dataset
df = pd.read_csv('flipkart.csv')
```

```
# Check for null values
print(df.isnull().sum())

# Handle null values (e.g., drop or fill)
df.dropna(inplace=True)  # or use df.fillna() if appropriate
```

```
Unnamed: 0      0
Product_name    0
Review          0
Rating          0
dtype: int64
```

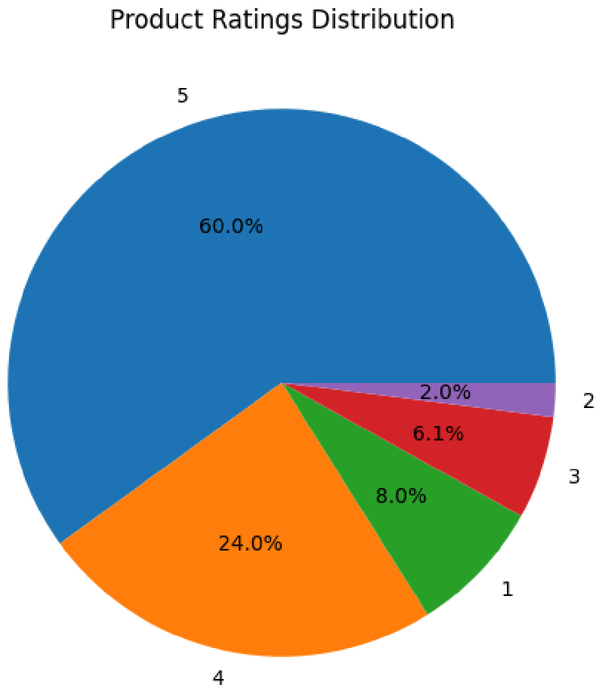
```
df.shape
```

```
(2304, 4)
```

```
df.columns
```

```
Index(['Unnamed: 0', 'Product_name', 'Review', 'Rating'], dtype='object')
```

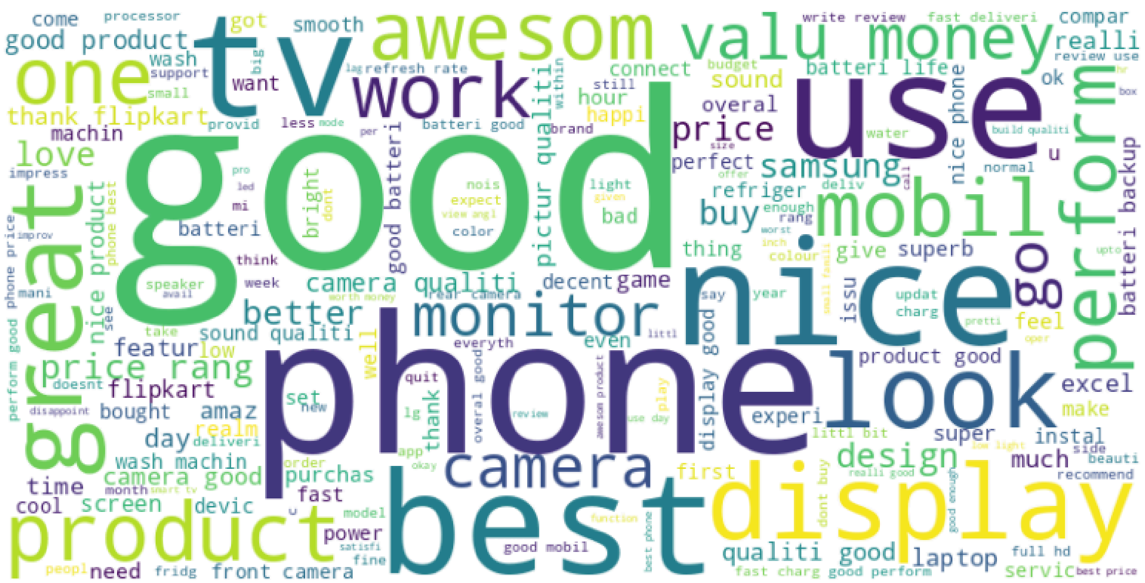
```
# Plot ratings distribution
plt.figure(figsize=(8, 6))
df['Rating'].value_counts().plot(kind='pie', autopct='%1.1f%%')
plt.title('Product Ratings Distribution')
plt.ylabel('')
plt.show()
```



```
import re
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
import nltk
nltk.download('punkt')

# Initialize stemmer and stopwords
stemmer = PorterStemmer()
stop_words = set(stopwords.words('english'))

# Function to clean text
def clean_text(text):
    text = text.lower()
    text = re.sub(r'\d+', '', text)
    text = re.sub(r'[^\w\s]', '', text)
    words = word_tokenize(text)
    words = [stemmer.stem(word) for word in words if word not in stop_words]
    return ' '.join(words)

df['cleaned_reviews'] = df['Review'].apply(clean_text)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

```
# Generate WordCloud
text = ' '.join(df['cleaned_reviews'])
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)

plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```



```
# Initialize sentiment analyzer
analyzer = SentimentIntensityAnalyzer()

# Function to get sentiment
def get_sentiment(text):
    score = analyzer.polarity_scores(text)
    return score

# Apply sentiment analysis
df['sentiment'] = df['cleaned_reviews'].apply(get_sentiment)
df['positive'] = df['sentiment'].apply(lambda x: x['pos'])
df['negative'] = df['sentiment'].apply(lambda x: x['neg'])
df['neutral'] = df['sentiment'].apply(lambda x: x['neu'])
```

```
# Calculate overall sentiment scores
overall_sentiment = df[['positive', 'negative', 'neutral']].mean()
print("Overall Sentiment Scores:")
print(overall_sentiment)
```

```
Overall Sentiment Scores:
```