In [6]:
```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\sandy\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[6]: True

59. Tokenization [Sentence & Word]

In [7]:
```
from nltk.tokenize import sent_tokenize
from nltk.tokenize import word_tokenize

f=open("da.txt")
text=f.read()
print(text)

sent=sent_tokenize(text)
print("Number of sentences:",len(sent))
for i in range(len(sent)):
    print("\nSentence:",i+1,"\n",sent[i])

w=word_tokenize(text)
print("\nTotal Words:",len(w))
print(w)
```

```
A smartphone is a cellular telephone with an integrated computer and other features not originally associated wi
th telephones, such as an operating system (OS), web browsing and the ability to run software applications. Smar
tphones are used by consumers and as part of a person's business or work.  Smartphones have become a very import
ant form of communication these days.
Number of sentences: 3

Sentence: 1
 A smartphone is a cellular telephone with an integrated computer and other features not originally associated w
ith telephones, such as an operating system (OS), web browsing and the ability to run software applications.

Sentence: 2
 Smartphones are used by consumers and as part of a person's business or work.

Sentence: 3
 Smartphones have become a very important form of communication these days.

Total Words: 66
['A', 'smartphone', 'is', 'a', 'cellular', 'telephone', 'with', 'an', 'integrated', 'computer', 'and', 'other',
'features', 'not', 'originally', 'associated', 'with', 'telephones', ',', 'such', 'as', 'an', 'operating', 'syst
em', '(', 'OS', ')', ',', 'web', 'browsing', 'and', 'the', 'ability', 'to', 'run', 'software', 'applications', '
.', 'Smartphones', 'are', 'used', 'by', 'consumers', 'and', 'as', 'part', 'of', 'a', 'person', "'s", 'business',
'or', 'work', '.', 'Smartphones', 'have', 'become', 'a', 'very', 'important', 'form', 'of', 'communication', 'th
ese', 'days', '.']
```

60. N-Grams

In [12]:
```
from nltk.util import ngrams
from nltk.tokenize import word_tokenize

f=open("da.txt")
text=f.read()
w=word_tokenize(text)

print("Bi-Grams:\n\n",list(ngrams(w,2)))
print("\n\nTri-Grams:\n\n",list(ngrams(w,3)))
```

Bi-Grams:

```
 [('A', 'smartphone'), ('smartphone', 'is'), ('is', 'a'), ('a', 'cellular'), ('cellular', 'telephone'), ('teleph
one', 'with'), ('with', 'an'), ('an', 'integrated'), ('integrated', 'computer'), ('computer', 'and'), ('and', 'o
ther'), ('other', 'features'), ('features', 'not'), ('not', 'originally'), ('originally', 'associated'), ('assoc
iated', 'with'), ('with', 'telephones'), ('telephones', ','), (',', 'such'), ('such', 'as'), ('as', 'an'), ('an'
, 'operating'), ('operating', 'system'), ('system', '('), ('(', 'OS'), ('OS', ')'), (')', ','), (',', 'web'), ('
web', 'browsing'), ('browsing', 'and'), ('and', 'the'), ('the', 'ability'), ('ability', 'to'), ('to', 'run'), ('
run', 'software'), ('software', 'applications'), ('applications', '.'), ('.', 'Smartphones'), ('Smartphones', 'a
re'), ('are', 'used'), ('used', 'by'), ('by', 'consumers'), ('consumers', 'and'), ('and', 'as'), ('as', 'part'),
('part', 'of'), ('of', 'a'), ('a', 'person'), ('person', "'s"), ("'s", 'business'), ('business', 'or'), ('or', '
work'), ('work', '.'), ('.', 'Smartphones'), ('Smartphones', 'have'), ('have', 'become'), ('become', 'a'), ('a',
'very'), ('very', 'important'), ('important', 'form'), ('form', 'of'), ('of', 'communication'), ('communication'
, 'these'), ('these', 'days'), ('days', '.')]
```

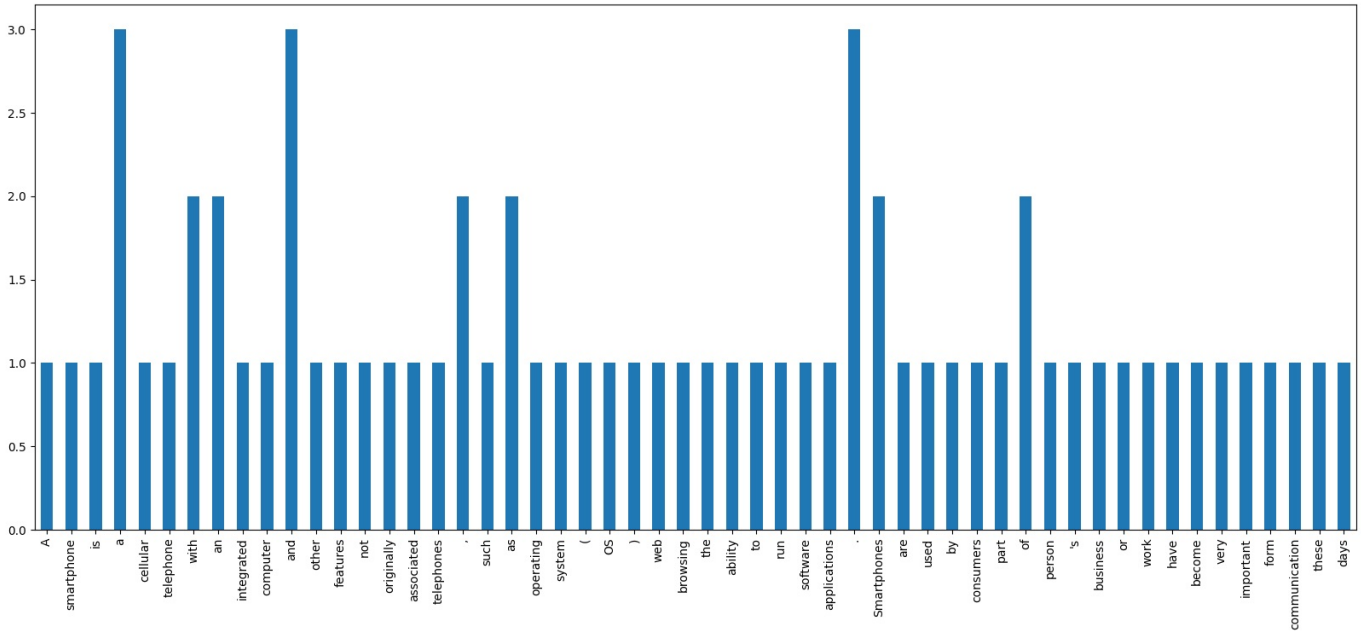
Tri-Grams:

```
 [('A', 'smartphone', 'is'), ('smartphone', 'is', 'a'), ('is', 'a', 'cellular'), ('a', 'cellular', 'telephone'),
('cellular', 'telephone', 'with'), ('telephone', 'with', 'an'), ('with', 'an', 'integrated'), ('an', 'integrated
', 'computer'), ('integrated', 'computer', 'and'), ('computer', 'and', 'other'), ('and', 'other', 'features'), (
'other', 'features', 'not'), ('features', 'not', 'originally'), ('not', 'originally', 'associated'), ('originall
y', 'associated', 'with'), ('associated', 'with', 'telephones'), ('with', 'telephones', ','), ('telephones', ','
, 'such'), (',', 'such', 'as'), ('such', 'as', 'an'), ('as', 'an', 'operating'), ('an', 'operating', 'system'),
('operating', 'system', '('), ('system', '(', 'OS'), ('(', 'OS', ')'), ('OS', ')', ','), (')', ',', 'web'), (','
, 'web', 'browsing'), ('web', 'browsing', 'and'), ('browsing', 'and', 'the'), ('and', 'the', 'ability'), ('the',
'ability', 'to'), ('ability', 'to', 'run'), ('to', 'run', 'software'), ('run', 'software', 'applications'), ('so
ftware', 'applications', '.'), ('applications', '.', 'Smartphones'), ('.', 'Smartphones', 'are'), ('Smartphones'
, 'are', 'used'), ('are', 'used', 'by'), ('used', 'by', 'consumers'), ('by', 'consumers', 'and'), ('consumers',
'and', 'as'), ('and', 'as', 'part'), ('as', 'part', 'of'), ('part', 'of', 'a'), ('of', 'a', 'person'), ('a', 'pe
rson', "'s"), ('person', "'s", 'business'), ("'s", 'business', 'or'), ('business', 'or', 'work'), ('or', 'work',
'.'), ('work', '.', 'Smartphones'), ('.', 'Smartphones', 'have'), ('Smartphones', 'have', 'become'), ('have', 'b
ecome', 'a'), ('become', 'a', 'very'), ('a', 'very', 'important'), ('very', 'important', 'form'), ('important',
'form', 'of'), ('form', 'of', 'communication'), ('of', 'communication', 'these'), ('communication', 'these', 'da
ys'), ('these', 'days', '.')]
```

61. Frequency Distribution of Words

In [13]:
```
import pandas as p
import matplotlib.pyplot as m
from nltk.probability import FreqDist
from nltk.tokenize import word_tokenize
f=open("da.txt")
text=f.read()
print(text)


w=word_tokenize(text)
freq=FreqDist(w)
print("Count of and word:",freq['and'])

freq=p.Series(dict(freq))
m.figure(figsize=(20,8))
freq.plot(kind='bar')
m.show()
```

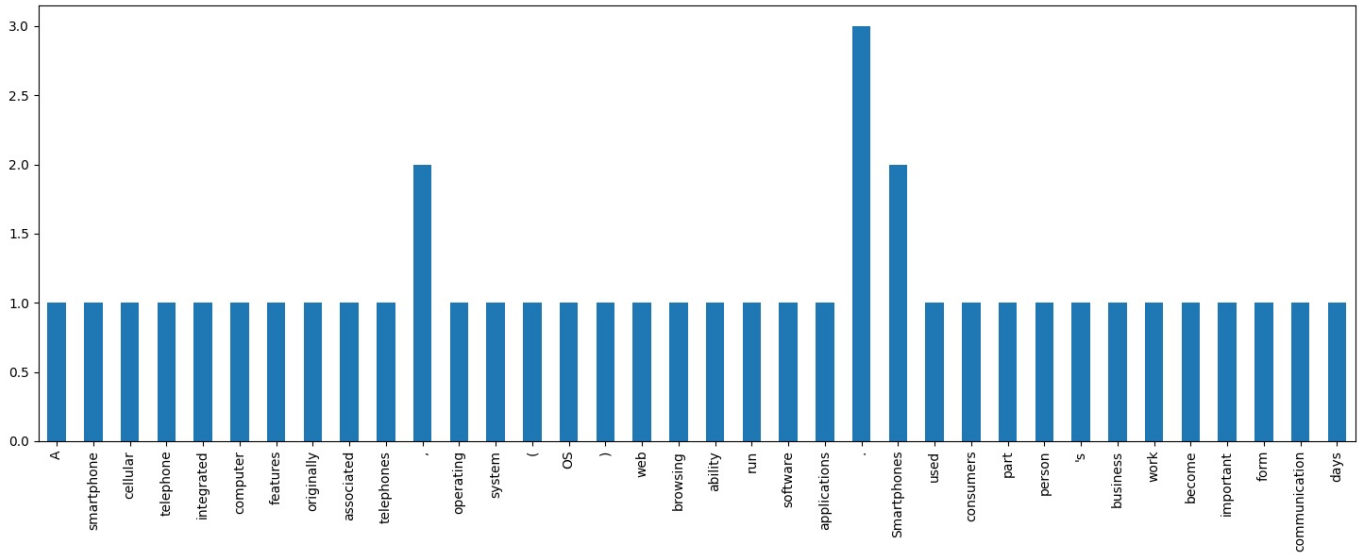
```
A smartphone is a cellular telephone with an integrated computer and other features not originally associated wi
th telephones, such as an operating system (OS), web browsing and the ability to run software applications. Smar
tphones are used by consumers and as part of a person's business or work.  Smartphones have become a very import
ant form of communication these days.
Count of and word: 3
```



62. Removing Stop-words.

In [15]:
```
import nltk
import matplotlib.pyplot as m
import pandas as p
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist
w = word_tokenize(text)
print("Before Removal of Stopwords words count:",len(w))
stop_w = nltk.corpus.stopwords.words('english')
removed_stopw = []
for i in w:
    if i not in stop_w:
        removed_stopw.append(i)
print("\nAfter Removal of StopWords:",len(removed_stopw))
freq = FreqDist(removed_stopw)
freq_series = p.Series(dict(freq))
m.figure(figsize=(18, 6))
freq_series.plot(kind='bar')
m.show()
```

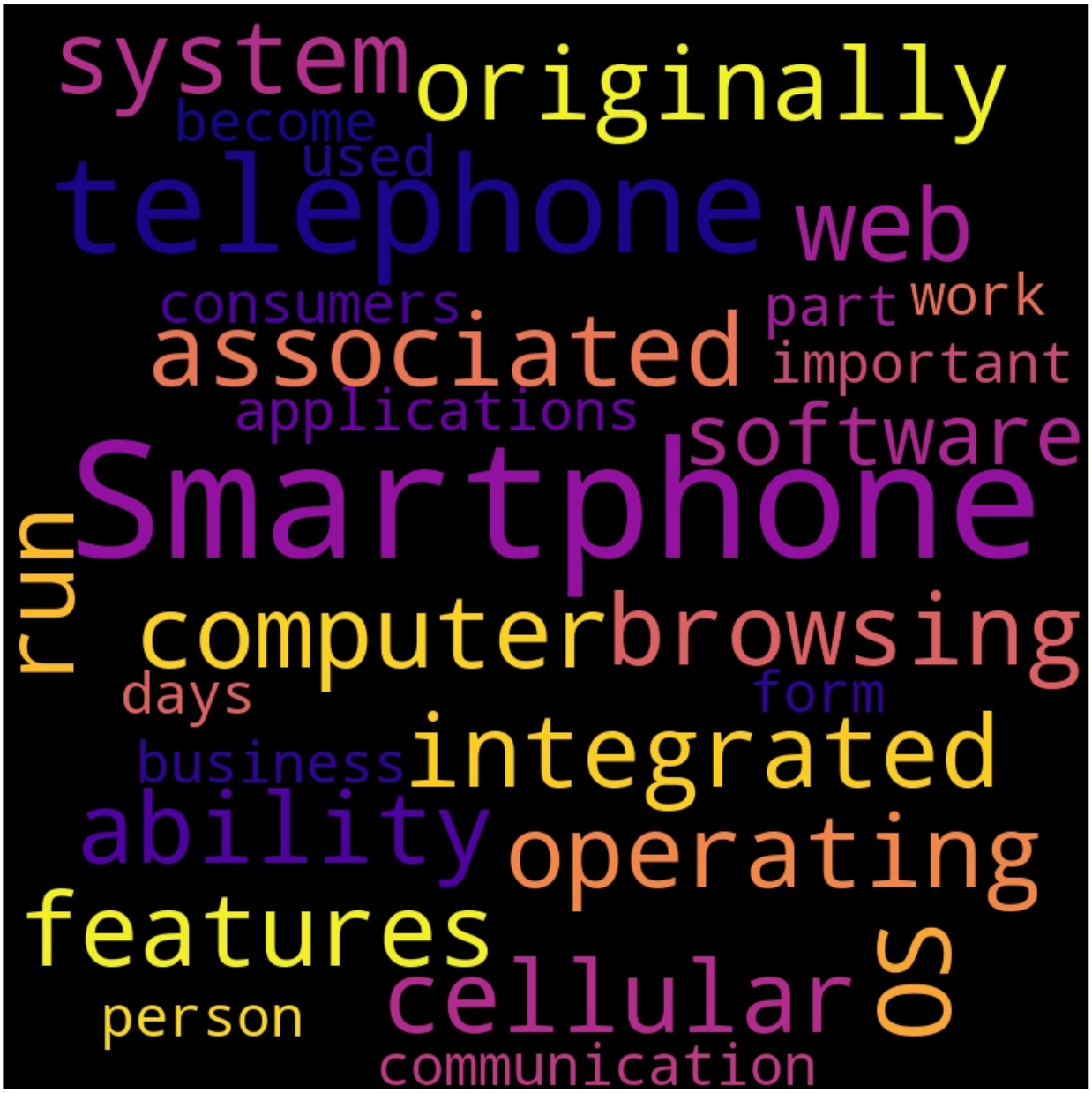
```
Before Removal of Stopwords words count: 66

After Removal of StopWords: 40
```



63. Word-cloud

In [20]:

```
import matplotlib.pyplot as m
f=open("da.txt")
text=f.read()
from wordcloud import WordCloud, STOPWORDS
stop_w=set(STOPWORDS)
wc=WordCloud(width=800,height=800,
                background_color='black',colormap='plasma',
                stopwords=stop_w,
                min_font_size=10).generate(text)
m.figure(figsize=(8,5),facecolor=None)
m.imshow(wc)
m.axis('off')
m.tight_layout(pad=0)
m.show()
```



64. Stemming & Lemmatization

In [8]:
```
import re
from nltk.tokenize import word_tokenize

f=open("da.txt")
text=f.read()
text=text.lower()

text=re.sub('[^A-Za-z0-9]+',' ',text)
text=re.sub("\S*\d\S*","",text).strip()
print(text)

w=word_tokenize(text,preserve_line=True)

from nltk.stem import PorterStemmer
ps=PorterStemmer()
ps_st=[ps.stem(i) for i in w]
print("\nStemming:\n\n",ps_st)

from nltk import WordNetLemmatizer
wnl=WordNetLemmatizer()
lema=[wnl.lemmatize(u) for u in w]

print("\n Lemmatization:\n\n",lema)
```

```
<>:9: SyntaxWarning: invalid escape sequence '\S'
<>:9: SyntaxWarning: invalid escape sequence '\S'
C:\Users\sandy\AppData\Local\Temp\ipykernel_14012\2729690726.py:9: SyntaxWarning: invalid escape sequence '\S'
  text=re.sub("\S*\d\S*","",text).strip()
```

```
a smartphone is a cellular telephone with an integrated computer and other features not originally associated wi
th telephones such as an operating system os web browsing and the ability to run software applications smartphon

es are used by consumers and as part of a person s business or work smartphones have become a very important for
m of communication these days

Stemming:

 ['a', 'smartphon', 'is', 'a', 'cellular', 'telephon', 'with', 'an', 'integr', 'comput', 'and', 'other', 'featur
', 'not', 'origin', 'associ', 'with', 'telephon', 'such', 'as', 'an', 'oper', 'system', 'os', 'web', 'brows', 'a
nd', 'the', 'abil', 'to', 'run', 'softwar', 'applic', 'smartphon', 'are', 'use', 'by', 'consum', 'and', 'as', 'p
art', 'of', 'a', 'person', 's', 'busi', 'or', 'work', 'smartphon', 'have', 'becom', 'a', 'veri', 'import', 'form
', 'of', 'commun', 'these', 'day']

 Lemmatization:

 ['a', 'smartphone', 'is', 'a', 'cellular', 'telephone', 'with', 'an', 'integrated', 'computer', 'and', 'other',
'feature', 'not', 'originally', 'associated', 'with', 'telephone', 'such', 'a', 'an', 'operating', 'system', 'o'
, 'web', 'browsing', 'and', 'the', 'ability', 'to', 'run', 'software', 'application', 'smartphones', 'are', 'use
d', 'by', 'consumer', 'and', 'a', 'part', 'of', 'a', 'person', 's', 'business', 'or', 'work', 'smartphones', 'ha
ve', 'become', 'a', 'very', 'important', 'form', 'of', 'communication', 'these', 'day']
```

In [24]:
```
import re
from nltk.tokenize import word_tokenize

f=open("da.txt")
text=f.read()
text=text.lower()

text=re.sub('[^A-Za-z0-9]+',' ',text)
text=re.sub(r"\S*\d\S*","",text).strip()
print(text)

w=word_tokenize(text,preserve_line=True)

from nltk.stem import PorterStemmer
ps=PorterStemmer()
ps_st=[ps.stem(i) for i in w]
print("\nStemming:\n\n",ps_st)
```

```
a smartphone is a cellular telephone with an integrated computer and other features not originally associated wi
th telephones such as an operating system os web browsing and the ability to run software applications smartphon
es are used by consumers and as part of a person s business or work smartphones have become a very important for
m of communication these days

Stemming:

 ['a', 'smartphon', 'is', 'a', 'cellular', 'telephon', 'with', 'an', 'integr', 'comput', 'and', 'other', 'featur
', 'not', 'origin', 'associ', 'with', 'telephon', 'such', 'as', 'an', 'oper', 'system', 'os', 'web', 'brows', 'a
nd', 'the', 'abil', 'to', 'run', 'softwar', 'applic', 'smartphon', 'are', 'use', 'by', 'consum', 'and', 'as', 'p
art', 'of', 'a', 'person', 's', 'busi', 'or', 'work', 'smartphon', 'have', 'becom', 'a', 'veri', 'import', 'form
', 'of', 'commun', 'these', 'day']
```

In [14]:
```
from nltk.stem import PorterStemmer

# create an object of class PorterStemmer
porter = PorterStemmer()
print(porter.stem("play"))
print(porter.stem("playing"))
print(porter.stem("plays"))
print(porter.stem("played"))
```

```
play
play
play
play
```

In [18]:
```
from nltk.stem import PorterStemmer
# create an object of class PorterStemmer
porter = PorterStemmer()
print(porter.stem("Communication"))
```

```
commun
```

In [22]:
```
import re
from nltk.tokenize import word_tokenize

f=open("da.txt")
text=f.read()
text=text.lower()

text=re.sub('[^A-Za-z0-9]+',' ',text)
text=re.sub(r"\S*\d\S*","",text).strip()
print(text)
```

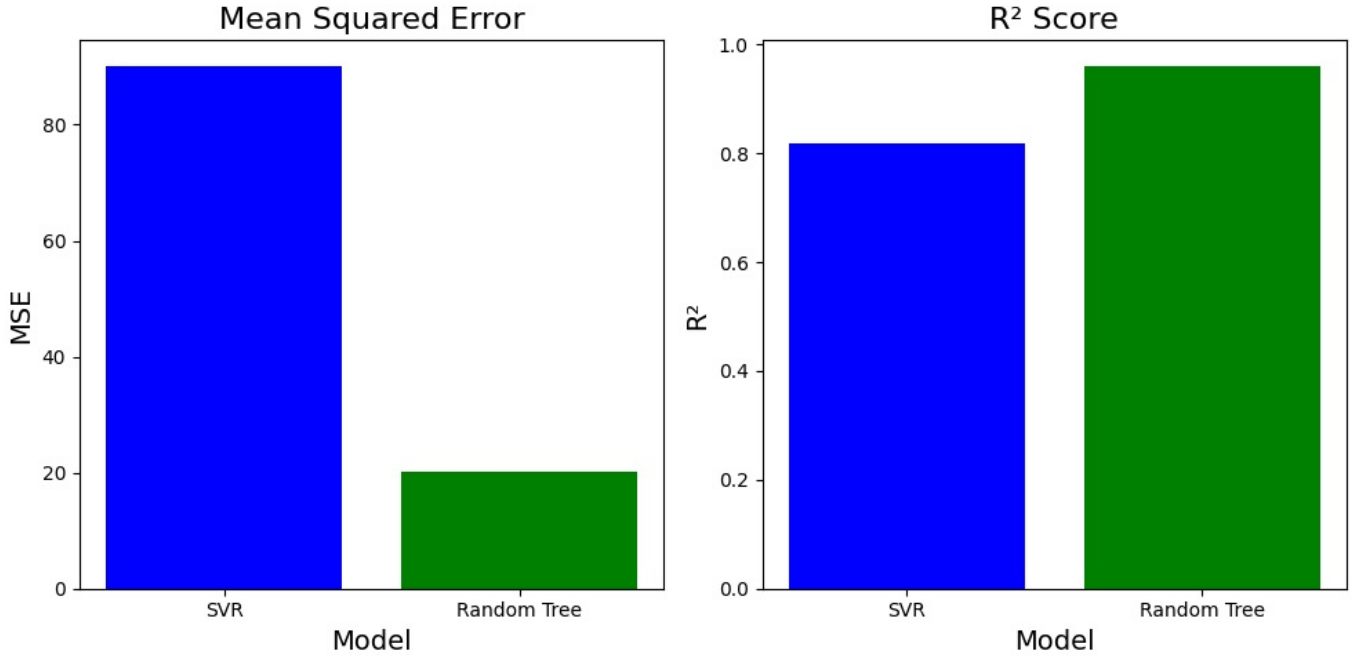
```
w=word_tokenize(text,preserve_line=True)

from nltk import WordNetLemmatizer
wnl=WordNetLemmatizer()
lema=[wnl.lemmatize(u) for u in w]
print("\n Lemmatization:\n\n",lema)
```

```
a smartphone is a cellular telephone with an integrated computer and other features not originally associated wi
th telephones such as an operating system os web browsing and the ability to run software applications smartphon
es are used by consumers and as part of a person s business or work smartphones have become a very important for
m of communication these days

 Lemmatization:

 ['a', 'smartphone', 'is', 'a', 'cellular', 'telephone', 'with', 'an', 'integrated', 'computer', 'and', 'other',
'feature', 'not', 'originally', 'associated', 'with', 'telephone', 'such', 'a', 'an', 'operating', 'system', 'o'
, 'web', 'browsing', 'and', 'the', 'ability', 'to', 'run', 'software', 'application', 'smartphones', 'are', 'use
d', 'by', 'consumer', 'and', 'a', 'part', 'of', 'a', 'person', 's', 'business', 'or', 'work', 'smartphones', 'ha
ve', 'become', 'a', 'very', 'important', 'form', 'of', 'communication', 'these', 'day']
```

In [28]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np
data = pd.read_csv('pd.csv')
data_cleaned = data.dropna()
X = data_cleaned[['Hours']].values
y = data_cleaned['Scores'].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
svr = SVR(kernel='rbf')
svr.fit(X_train_scaled, y_train)
y_pred_svr = svr.predict(X_test_scaled)

tree = DecisionTreeRegressor(random_state=42)
tree.fit(X_train, y_train)
y_pred_tree = tree.predict(X_test)

# Step 8: Evaluate the models
svr_mse = mean_squared_error(y_test, y_pred_svr)
svr_r2 = r2_score(y_test, y_pred_svr)

tree_mse = mean_squared_error(y_test, y_pred_tree)
tree_r2 = r2_score(y_test, y_pred_tree)

# Step 9: Print the results
print(f"Support Vector Regression (SVR) - MSE: {svr_mse:.2f}, R²: {svr_r2:.2f}")
print(f"Random Decision Tree Regressor - MSE: {tree_mse:.2f}, R²: {tree_r2:.2f}")

# Step 10: (Optional) Plot the results
import matplotlib.pyplot as plt

models = ['SVR', 'Random Tree']
mse_values = [svr_mse, tree_mse]
r2_values = [svr_r2, tree_r2]

print("train set:",X_train.shape,y_train.shape)
print("test set:",X_test.shape,y_test.shape)

# MSE plot
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.bar(models, mse_values, color=['blue', 'green'])
plt.title('Mean Squared Error', fontsize=16)
plt.xlabel('Model', fontsize=14)
plt.ylabel('MSE', fontsize=14)

# R² plot
plt.subplot(1, 2, 2)
plt.bar(models, r2_values, color=['blue', 'green'])
plt.title('R² Score', fontsize=16)
plt.xlabel('Model', fontsize=14)
plt.ylabel('R²', fontsize=14)
s
plt.tight_layout()
plt.show()
```

```
Support Vector Regression (SVR) - MSE: 90.05, R²: 0.82
Random Decision Tree Regressor - MSE: 20.10, R²: 0.96
train set: (102, 1) (102,)
test set: (44, 1) (44,)
```



In [ ]: