Analysing job requirements with NLP

In this phase I am going to use various NLP techniques to analyse the job descriptions from linkedin. The objective here is to get the idea about what words and phrases exactly should be in the resume so that the initial screening is cleared.

In [80]:
```python
#getting the packages
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import pandas as pd
import string
import re
from sklearn.cluster import KMeans
from nltk.stem import WordNetLemmatizer
from nltk.corpus import wordnet
import matplotlib.pyplot as plt
from sklearn.metrics import silhouette_score, calinski_harabasz_score
from wordcloud import WordCloud
import pickle
```

In [81]:
```python
#downloading the file
lndf = pd.read_csv("C:\\Users\\sujoydutta\\Desktop\\Data analysis\\Datasets for ML\\NLP\\job_descriptions_linke
lndf.head()
```

Out[81]:

	job description
0	About the job\nThis role has been designated a...
1	About the job\nABOUT US:\n\nBraintrust is a us...
2	About the job\nAt Snaphunt, we believe that th...
3	About the job\nAmerican Express invites you to...
4	About the job\nJob Descriptions:- Promanage It...

In [82]:
```python
#viewing the dataset
lndf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 1 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   job description  100 non-null    object
dtypes: object(1)
memory usage: 928.0+ bytes
```

In [83]:
```python
data_science_corpus = [
    "data science",
    "data",
    "Python",
    "powerbi",
    'python programming',
    "Power BI",
    "machine learning",
    "artificial intelligence",
    "data analysis",
    "data analytics",
    "data scientist",
    "data engineer",
    "data analyst",
    "data mining",
    "data visualization",
    "predictive modeling",
    "statistical modeling",
    "supervised learning",
    "unsupervised learning",
    "deep learning",
    "neural networks",
    "natural language processing",
    "reinforcement learning",
    "ensemble learning",
    "feature engineering",
    "model training",
    "model evaluation",
    "model deployment",
    "data preprocessing",
    "data cleaning",
    "data wrangling",
    "feature selection",
    "dimensionality reduction",
    "cross-validation",
    "hyperparameter tuning",
    "overfitting",
    "underfitting",
    "bias-variance tradeoff",
    "accuracy",
    "precision",
    "recall",
    "f1-score",
    "confusion matrix",
    "ROC curve",
    "AUC-ROC",
    "gradient descent",
    "random forest",
    "decision tree",
    "support vector machine",
    "k-nearest neighbors",
    "clustering",
    "classification",
    "regression",
    "evaluation metrics",
    "model interpretation",
    "model explainability",
    "model performance",
    "model selection",
    "model optimization",
    "model deployment",
    "data-driven",
    "analytical",
    "problem-solving",
    "critical thinking",
    "detail-oriented",
    "communication",
    "collaboration",
    "teamwork",
    "leadership",
    "adaptability",
    'R',
    'tableau',
    'qlikview',
    "continuous learning",
    "creativity",
    "innovation",
    "strategic thinking",
    "business acumen",
    "presentation skills",
    "project management",
    "time management",
    "self-motivation",
    "multitasking",
    "organizational skills",
    "attention to detail",
    "technical skills",
    "quantitative skills",
    "qualitative skills"
]
```

In [84]:
```python
# Define the corpus of verbs and adjectives related to data science
data_science_verbs_adjectives = [
    "analyze",
    "predict",
    "solve",
    "optimize",
    "model",
    "visualize",
    "extract",
    "evaluate",
    "deploy",
    "preprocess",
    "explore",
    "wrangle",
    "interpret",
    "identify",
    "collect",
    "consolidate",
    "diligence",
    "design",
    "build",
    "track",
    "assess",
    "execute",
    "interpret",
    "efficient",
    "advanced",
    "deep",
    "complex",
    "statistical",
    "predictive",
    "analytical",
    "visual",
    "real-time",
    "algorithmic",
    "technical",
    "machine",
    "artificial",
    "neural",
    "quantitative",
    "qualitative",
    "structured",
    "unstructured"
]
```

In [85]:
```python
# Define the corpus of technology
techcorpus = [
    'Python',
    'R',
    'SQL',
    'Java',
    'Scala',
    'MATLAB',
    'Tableau',
    'Power BI',
    'Excel',
    'Pandas',
    'NumPy',
    'SciPy',
    'scikit-learn',
    'TensorFlow',
    'Keras',
    'PyTorch',
    'NLTK',
    'Natural Language Processing',
    'OpenCV',
    'Apache Spark',
    'Hadoop',
    'AWS',
    'Azure',
    'Google Cloud',
    'Docker',
    'Git',
    'Jupyter Notebook',
    'Visual Studio Code',
    'Sublime Text',
    'Dataiku',
    'Knime',
    'Alteryx',
    'BigQuery',
    'Redshift',
    'MySQL',
    'PostgreSQL',
    'MongoDB',
    'Cassandra',
    'Hive',
    'Apache Kafka',
    'Apache Airflow',
    'Apache Beam',
    'Dask',
    'Streamlit',
    'Flask',
    'Django',
    'Plotly',
    'Seaborn',
    'Matplotlib',
    'Bokeh',
    'ggplot',
    'StatsModels',
    'XGBoost',
    'LightGBM',
    'CatBoost',
    'SVM',
    'Random Forest',
    'Linear Regression',
    'Logistic Regression',
    'K-means',
    'PCA',
    'Reinforcement Learning',
    'Time Series Analysis',
    'Deep Learning',
    'Neural Networks',
    'Data Mining',
    'Data Wrangling',
    'Feature Engineering',
    'Model Evaluation',
    'Model Deployment',
    'Data Cleaning',
    'Data Visualization',
    'Exploratory Data Analysis',
    'Machine Learning',
    'Artificial Intelligence',
    'Predictive Analytics',
    'Statistical Analysis',
    'Text Mining',
    'Image Processing',
    'Web Scraping',
    'Version Control',
    'Collaborative Filtering',
    'Recommendation Systems',
    'Anomaly Detection',
    'Dimensionality Reduction',
    'Ensemble Learning',
    'Model Selection',
    'Cross-Validation',
    'Hyperparameter Tuning',
    'Model Interpretability',
    'Model Metrics',
    'Feature Selection',
    'Data Pipelines',
    'Streaming Data',
    'Data Engineering',
    'Model Deployment',
    'Cloud Computing',
    'Containerization',
    'API Development',
    'Web Development',
    'Time Series Forecasting',
    'Unsupervised Learning',
    'Supervised Learning',
    'Natural Language Understanding',
    'Computer Vision',
    'Parallel Computing',
    'Data Privacy',
    'Data Governance',
    'Data Security',
    'Data Ethics',
    'Data Quality',
    'Data Integration',
    'Data Architecture',
    'ETL',
    'Big Data',
    'Internet of Things',
    'Blockchain',
    'Cybersecurity'
]
```

In [86]:
```python
#qualities of data analyst/scientist
quality_corpus = [
    'Analytical',
    'Problem-solving',
    'Detail-oriented',
    'Critical thinking',
    'Curious',
    'Logical',
    'Creative',
    'Inquisitive',
    'Attention to detail',
    'Mathematical',
    'Statistical',
    'Technical',
    'Business acumen',
    'Domain knowledge',
    'Effective communication',
    'Data-driven',
    'Research-oriented',
    'Adaptable',
    'Collaborative',
    'Team player',
    'Time management',
    'Project management',
    'Organized',
    'Results-oriented',
    'Self-motivated',
    'Continuous learning',
    'Data interpretation',
    'Data analysis',
    'Data visualization',
    'Data modeling',
    'Data mining',
    'Data cleansing',
    'Data wrangling',
    'Data storytelling',
    'SQL querying',
    'Python programming',
    'R programming',
    'Machine learning',
    'Predictive modeling',
    'Statistical analysis',
    'Reporting',
    'Dashboarding',
    'Data governance',
    'Data privacy',
    'Data security',
    'Data ethics',
    'Data quality assurance',
    'ETL (Extract, Transform, Load)',
    'Data warehouse management',
    'Database querying',
    'Data integration',
    'Data transformation',
    'Data validation',
    'Data manipulation',
    'Data insights',
    'Data strategy',
    'Business intelligence',
    'Visualization tools',
    'Statistical techniques',
    'Machine learning algorithms',
    'Data-driven decision-making',
    'Collaboration tools',
    'Presentation skills',
    'Client interaction',
    'Problem identification',
    'Solution development',
    'Data governance frameworks',
    'Quality assurance',
    'Data architecture',
    'Data documentation',
    'Data ethics compliance',
    'Data security measures',
    'Data privacy regulations',
    'Data strategy planning',
    'Data visualization techniques',
    'Effective reporting',
    'Business understanding',
    'Domain-specific knowledge',
    'Data analysis best practices',
    'Continuous improvement',
    'Data insights communication',
    'Strategic thinking',
    'Innovation mindset',
    'Stakeholder management',
    'Risk assessment',
    'Data-driven decision support',
    'Data storytelling expertise',
    'Data presentation skills',
    'Data project management',
    'Data governance policies',
    'Data privacy frameworks',
    'Data security protocols',
    'Data quality management',
    'Data integration methods',
    'Data transformation strategies',
    'Data validation techniques',
    'Data manipulation skills',
    'Data warehouse design',
    'SQL optimization',
    'Programming proficiency',
    'Modeling techniques',
    'Algorithm implementation',
    'Statistical hypothesis testing',
    'Visualization platforms',
    'Business intelligence tools',
    'Data exploration methods',
    'Data interpretation frameworks',
    'Data reporting methodologies',
    'Data collaboration platforms',
    'Project management methodologies'
]
```

In [87]:
```python
quality_corpus = [
    'Analytical',
    'Problem-solving',
    'Detail-oriented',
    'Critical thinking',
    'Curious',
    'Logical',
    'Creative',
    'Inquisitive',
    'Attention to detail',
    'Mathematical',
    'Statistical',
    'Technical',
    'Business acumen',
    'Domain knowledge',
    'Effective communication',
    'Data-driven',
    'Research-oriented',
    'Adaptable',
    'Collaborative',
    'Team player',
    'Time management',
    'Project management',
    'Organized',
    'Results-oriented',
    'Self-motivated',
    'Continuous learning',
    'Data interpretation',
    'Data analysis',
    'Data visualization',
    'Data modeling',
    'Data mining',
    'Data cleansing',
    'Data wrangling',
    'Data storytelling',
    'SQL querying',
    'Python programming',
    'R programming',
    'Machine learning',
    'Predictive modeling',
    'Statistical analysis',
    'Reporting',
    'Dashboarding',
    'Data governance',
    'Data privacy',
    'Data security',
    'Data ethics',
    'Data quality assurance',
    'ETL (Extract, Transform, Load)',
    'Data warehouse management',
    'Database querying',
    'Data integration',
    'Data transformation',
    'Data validation',
    'Data manipulation',
    'Data insights',
    'Data strategy',
    'Business intelligence',
    'Visualization tools',
    'Statistical techniques',
    'Machine learning algorithms',
    'Data-driven decision-making',
    'Collaboration tools',
    'Presentation skills',
    'Client interaction',
    'Problem identification',
    'Solution development',
    'Data governance frameworks',
    'Quality assurance',
    'Data architecture',
    'Data documentation',
    'Data ethics compliance',
    'Data security measures',
    'Data privacy regulations',
    'Data strategy planning',
    'Data visualization techniques',
    'Effective reporting',
    'Business understanding',
    'Domain-specific knowledge',
    'Data analysis best practices',
    'Continuous improvement',
    'Data insights communication',
    'Strategic thinking',
    'Innovation mindset',
    'Stakeholder management',
    'Risk assessment',
    'Data-driven decision support',
    'Data storytelling expertise',
    'Data presentation skills',
    'Data project management',
    'Data governance policies',
    'Data privacy frameworks',
    'Data security protocols',
    'Data quality management',
    'Data integration methods',
    'Data transformation strategies',
    'Data validation techniques',
    'Data manipulation skills',
    'Data warehouse design',
    'SQL optimization',
    'Programming proficiency',
    'Modeling techniques',
    'Algorithm implementation',
    'Statistical hypothesis testing',
    'Visualization platforms',
    'Business intelligence tools',
    'Data exploration methods',
    'Data interpretation frameworks',
    'Data reporting methodologies',
    'Data collaboration platforms',
    'Project management methodologies'
]
```

In [88]:
```python
#making a combined words
data_science_words=data_science_verbs_adjectives+data_science_corpus+quality_corpus+techcorpus
data_science_words
```

Out[88]:
```
['analyze',
 'predict',
 'solve',
 'optimize',
 'model',
 'visualize',
 'extract',
 'evaluate',
 'deploy',
 'preprocess',
 'explore',
 'wrangle',
 'interpret',
 'identify',
 'collect',
 'consolidate',
 'diligence',
 'design',
 'build',
 'track',
 'assess',
 'execute',
 'interpret',
 'efficient',
 'advanced',
 'deep',
 'complex',
 'statistical',
 'predictive',
 'analytical',
 'visual',
 'real-time',
 'algorithmic',
 'technical',
 'machine',
 'artificial',
 'neural',
 'quantitative',
 'qualitative',
 'structured',
 'unstructured',
 'data science',
 'data',
 'Python',
 'powerbi',
 'python programming',
 'Power BI',
 'machine learning',
 'artificial intelligence',
 'data analysis',
 'data analytics',
 'data scientist',
 'data engineer',
 'data analyst',
 'data mining',
 'data visualization',
 'predictive modeling',
 'statistical modeling',
 'supervised learning',
 'unsupervised learning',
 'deep learning',
 'neural networks',
 'natural language processing',
 'reinforcement learning',
 'ensemble learning',
 'feature engineering',
 'model training',
 'model evaluation',
 'model deployment',
 'data preprocessing',
 'data cleaning',
 'data wrangling',
 'feature selection',
 'dimensionality reduction',
 'cross-validation',
 'hyperparameter tuning',
 'overfitting',
 'underfitting',
 'bias-variance tradeoff',
 'accuracy',
 'precision',
 'recall',
 'f1-score',
 'confusion matrix',
 'ROC curve',
 'AUC-ROC',
 'gradient descent',
 'random forest',
 'decision tree',
 'support vector machine',
 'k-nearest neighbors',
 'clustering',
 'classification',
 'regression',
 'evaluation metrics',
 'model interpretation',
 'model explainability',
 'model performance',
 'model selection',
 'model optimization',
 'model deployment',
 'data-driven',
 'analytical',
 'problem-solving',
 'critical thinking',
 'detail-oriented',
 'communication',
 'collaboration',
 'teamwork',
 'leadership',
 'adaptability',
 'R',
 'tableau',
 'qlikview',
 'continuous learning',
 'creativity',
 'innovation',
 'strategic thinking',
 'business acumen',
 'presentation skills',
 'project management',
 'time management',
 'self-motivation',
 'multitasking',
 'organizational skills',
 'attention to detail',
 'technical skills',
 'quantitative skills',
 'qualitative skills',
 'Analytical',
 'Problem-solving',
 'Detail-oriented',
 'Critical thinking',
 'Curious',
 'Logical',
 'Creative',
 'Inquisitive',
 'Attention to detail',
 'Mathematical',
 'Statistical',
 'Technical',
 'Business acumen',
 'Domain knowledge',
 'Effective communication',
 'Data-driven',
 'Research-oriented',
 'Adaptable',
 'Collaborative',
 'Team player',
 'Time management',
 'Project management',
 'Organized',
 'Results-oriented',
 'Self-motivated',
 'Continuous learning',
 'Data interpretation',
 'Data analysis',
 'Data visualization',
 'Data modeling',
 'Data mining',
 'Data cleansing',
 'Data wrangling',
 'Data storytelling',
 'SQL querying',
 'Python programming',
 'R programming',
 'Machine learning',
 'Predictive modeling',
 'Statistical analysis',
 'Reporting',
 'Dashboarding',
 'Data governance',
 'Data privacy',
 'Data security',
 'Data ethics',
 'Data quality assurance',
 'ETL (Extract, Transform, Load)',
 'Data warehouse management',
 'Database querying',
 'Data integration',
 'Data transformation',
 'Data validation',
 'Data manipulation',
 'Data insights',
 'Data strategy',
 'Business intelligence',
 'Visualization tools',
 'Statistical techniques',
 'Machine learning algorithms',
 'Data-driven decision-making',
 'Collaboration tools',
 'Presentation skills',
 'Client interaction',
 'Problem identification',
 'Solution development',
 'Data governance frameworks',
 'Quality assurance',
 'Data architecture',
 'Data documentation',
 'Data ethics compliance',
 'Data security measures',
 'Data privacy regulations',
 'Data strategy planning',
 'Data visualization techniques',
 'Effective reporting',
 'Business understanding',
 'Domain-specific knowledge',
 'Data analysis best practices',
 'Continuous improvement',
 'Data insights communication',
 'Strategic thinking',
 'Innovation mindset',
 'Stakeholder management',
 'Risk assessment',
 'Data-driven decision support',
 'Data storytelling expertise',
 'Data presentation skills',
 'Data project management',
 'Data governance policies',
 'Data privacy frameworks',
 'Data security protocols',
 'Data quality management',
 'Data integration methods',
 'Data transformation strategies',
 'Data validation techniques',
 'Data manipulation skills',
 'Data warehouse design',
 'SQL optimization',
 'Programming proficiency',
 'Modeling techniques',
 'Algorithm implementation',
 'Statistical hypothesis testing',
 'Visualization platforms',
 'Business intelligence tools',
 'Data exploration methods',
 'Data interpretation frameworks',
 'Data reporting methodologies',
 'Data collaboration platforms',
 'Project management methodologies',
 'Python',
 'R',
 'SQL',
 'Java',
 'Scala',
 'MATLAB',
 'Tableau',
 'Power BI',
 'Excel',
 'Pandas',
 'NumPy',
 'SciPy',
 'scikit-learn',
 'TensorFlow',
 'Keras',
 'PyTorch',
 'NLTK',
 'Natural Language Processing',
 'OpenCV',
 'Apache Spark',
 'Hadoop',
 'AWS',
 'Azure',
 'Google Cloud',
 'Docker',
 'Git',
 'Jupyter Notebook',
 'Visual Studio Code',
 'Sublime Text',
 'Dataiku',
 'Knime',
 'Alteryx',
 'BigQuery',
 'Redshift',
 'MySQL',
 'PostgreSQL',
 'MongoDB',
 'Cassandra',
 'Hive',
 'Apache Kafka',
 'Apache Airflow',
 'Apache Beam',
 'Dask',
 'Streamlit',
 'Flask',
 'Django',
 'Plotly',
 'Seaborn',
 'Matplotlib',
 'Bokeh',
 'ggplot',
 'StatsModels',
 'XGBoost',
 'LightGBM',
 'CatBoost',
 'SVM',
 'Random Forest',
 'Linear Regression',
 'Logistic Regression',
 'K-means',
 'PCA',
 'Reinforcement Learning',
 'Time Series Analysis',
 'Deep Learning',
 'Neural Networks',
 'Data Mining',
 'Data Wrangling',
 'Feature Engineering',
 'Model Evaluation',
 'Model Deployment',
 'Data Cleaning',
 'Data Visualization',
 'Exploratory Data Analysis',
 'Machine Learning',
 'Artificial Intelligence',
 'Predictive Analytics',
 'Statistical Analysis',
 'Text Mining',
 'Image Processing',
 'Web Scraping',
 'Version Control',
 'Collaborative Filtering',
 'Recommendation Systems',
 'Anomaly Detection',
 'Dimensionality Reduction',
 'Ensemble Learning',
 'Model Selection',
 'Cross-Validation',
 'Hyperparameter Tuning',
 'Model Interpretability',
 'Model Metrics',
 'Feature Selection',
 'Data Pipelines',
 'Streaming Data',
 'Data Engineering',
 'Model Deployment',
 'Cloud Computing',
 'Containerization',
 'API Development',
 'Web Development',
 'Time Series Forecasting',
 'Unsupervised Learning',
 'Supervised Learning',
 'Natural Language Understanding',
 'Computer Vision',
 'Parallel Computing',
 'Data Privacy',
 'Data Governance',
 'Data Security',
 'Data Ethics',
 'Data Quality',
 'Data Integration',
 'Data Architecture',
 'ETL',
 'Big Data',
 'Internet of Things',
 'Blockchain',
 'Cybersecurity']
```

In [89]:
```python
# Remove punctuation and convert text to lowercase
lndf['job description'] = lndf['job description'].str.replace("[^\w\s\n]", "").str.lower()
```

```
C:\Users\sujoydutta\AppData\Local\Temp\ipykernel_1156\3138801493.py:2: FutureWarning: The default value of rege
x will change from True to False in a future version.
  lndf['job description'] = lndf['job description'].str.replace("[^\w\s\n]", "").str.lower()
```

In [90]:
```python
# Remove symbols from the text
lndf['job description'] = lndf['job description'].str.replace(r'[@#$%^()\[\]]', '')

# Print the updated job descriptions
lndf['job description']
```

```
C:\Users\sujoydutta\AppData\Local\Temp\ipykernel_1156\3202076043.py:2: FutureWarning: The default value of rege
x will change from True to False in a future version.
  lndf['job description'] = lndf['job description'].str.replace(r'[@#$%^()\[\]]', '')
```

Out[90]:
```
0     about the job\nthis role has been designated a...
1     about the job\nabout us\n\nbraintrust is a use...
2     about the job\nat snaphunt we believe that the...
3     about the job\namerican express invites you to...
4     about the job\njob descriptions promanage it s...
                            ...
95    about the job\njjopentester enterprise is a co...
96    about the job\ngreetings\n\n\ncontractual hiri...
97    about the job\nare you passionate about music ...
98    about the job\njob description\n\n\nexcellent ...
99    about the job\nquadeye is one of the largest i...
Name: job description, Length: 100, dtype: object
```

In [91]:
```python
# Tokenize the text
lndf['job description'] = lndf['job description'].apply(word_tokenize)
lndf['job description']
```

Out[91]:
```
0     [about, the, job, this, role, has, been, desig...
1     [about, the, job, about, us, braintrust, is, a...
2     [about, the, job, at, snaphunt, we, believe, t...
3     [about, the, job, american, express, invites, ...
4     [about, the, job, job, descriptions, promanage...
                            ...
95    [about, the, job, jjopentester, enterprise, is...
96    [about, the, job, greetings, contractual, hiri...
97    [about, the, job, are, you, passionate, about,...
98    [about, the, job, job, description, excellent,...
99    [about, the, job, quadeye, is, one, of, the, l...
Name: job description, Length: 100, dtype: object
```

In [92]:
```python
# Removing stopwords
stopwords_list = stopwords.words("english")
lndf['job description'] = lndf['job description'].apply(lambda tokens: [word for word in tokens if word not in
lndf['job description']
```

Out[92]:
```
0     [job, role, designated, âedgeâ, means, primari...
1     [job, us, braintrust, userowned, talent, netwo...
2     [job, snaphunt, believe, future, work, remote,...
3     [job, american, express, invites, share, resum...
4     [job, job, descriptions, promanage, solutions,...
                            ...
95    [job, jjopentester, enterprise, consultation, ...
96    [job, greetings, contractual, hiring, ernst, y...
97    [job, passionate, music, experience, working, ...
98    [job, job, description, excellent, spoken, wri...
99    [job, quadeye, one, largest, indiabased, propr...
Name: job description, Length: 100, dtype: object
```

In [93]:
```python
# Lemmatizing the text
lemmatizer = WordNetLemmatizer()
lndf['job description'] = lndf['job description'].apply(lambda tokens: [lemmatizer.lemmatize(word) for word in
lndf['job description']
```

Out[93]:
```
0     [job, role, designated, âedgeâ, mean, primaril...
1     [job, u, braintrust, userowned, talent, networ...
2     [job, snaphunt, believe, future, work, remote,...
3     [job, american, express, invite, share, resume...
4     [job, job, description, promanage, solution, l...
                            ...
95    [job, jjopentester, enterprise, consultation, ...
96    [job, greeting, contractual, hiring, ernst, yo...
97    [job, passionate, music, experience, working, ...
98    [job, job, description, excellent, spoken, wri...
99    [job, quadeye, one, largest, indiabased, propr...
Name: job description, Length: 100, dtype: object
```

In [94]:
```python
# Filter relevant words related to data science
lndf['job description'] = lndf['job description'].apply(lambda tokens: [word for word in tokens if word in data_

lndf['job description']
```

Out[94]:
```
0     [analyze, data, complex, extract, data, data, ...
1     [algorithmic, model, model, data, data, statis...
2     [predictive, model, data, statistical, data, m...
3     [data, leadership, design, collaboration, anal...
4     [technical, technical, optimize, optimize, opt...
                            ...
95    [predictive, model, data, statistical, data, m...
96    [communication, data, data, data, accuracy, an...
97    [optimize, analytical, data, communication, bu...
98     [communication, data, data, data, communication]
99    [track, quantitative, explore, quantitative, s...
Name: job description, Length: 100, dtype: object
```

In [95]:
```python
#checking the final dataset
lndf['job description']
```

Out[95]:
```
0     [analyze, data, complex, extract, data, data, ...
1     [algorithmic, model, model, data, data, statis...
2     [predictive, model, data, statistical, data, m...
3     [data, leadership, design, collaboration, anal...
4     [technical, technical, optimize, optimize, opt...
                            ...
95    [predictive, model, data, statistical, data, m...
96    [communication, data, data, data, accuracy, an...
97    [optimize, analytical, data, communication, bu...
98     [communication, data, data, data, communication]
99    [track, quantitative, explore, quantitative, s...
Name: job description, Length: 100, dtype: object
```

In [96]:
```python
# Convert the column to string type
lndf['job description'] = lndf['job description'].astype(str)

# Count distinct comma-separated words
word_counts = lndf['job description'].str.split(',').explode().str.strip().value_counts()

# Display the word counts
print(word_counts)
```

```
'data'              672
'communication'      69
'model'              67
'machine'            57
'analytical'         52
                   ...
'collect'             1
['identify'           1
'collaboration']      1
['algorithmic'        1
['design'             1
Name: job description, Length: 99, dtype: int64
```

In [97]:
```python
# Convert the column to string type
lndf['job description'] = lndf['job description'].astype(str)

# Count distinct comma-separated words
word_counts = lndf['job description'].str.split(',').explode().str.strip().value_counts()

# Separate the words and counts into variables
words = word_counts.index.tolist()
counts = word_counts.values.tolist()

# Display the words and their counts
for word, count in zip(words, counts):
    print(word, count)
```

```
'data' 672
'communication' 69
'model' 67
'machine' 57
'analytical' 52
'technical' 48
'build' 45
'identify' 44
'complex' 43
['data' 37
'advanced' 32
'statistical' 30
'data'] 30
'design' 28
'leadership' 25
'analyze' 23
'tableau' 23
'deep' 22
'quantitative' 22
'collaboration' 21
'predictive' 20
'innovation' 19
'communication'] 19
'accuracy' 17
'optimize' 16
'solve' 16
'execute' 10
'track' 10
'extract' 9
'evaluate' 8
'efficient' 7
['build' 7
'powerbi' 7
['communication' 6
['technical' 5
'explore' 5
['model' 5
'teamwork' 5
'structured' 5
['advanced' 5
'innovation'] 5
'creativity' 5
'artificial' 5
['innovation' 4
'clustering' 4
'model'] 4
'deploy' 4
'technical'] 4
'classification' 3
'regression' 3
['track' 3
'design'] 3
['analytical' 3
'machine'] 3
[] 3
'advanced'] 3
'visual' 3
'complex'] 3
'algorithmic' 3
'identify'] 3
['predictive' 3
'leadership'] 3
'interpret' 3
['optimize' 2
['deep' 2
'build'] 2
'unstructured' 2
['complex' 2
'track'] 2
'qualitative' 2
['machine' 1
['artificial' 1
['efficient' 1
'predictive'] 1
'neural' 1
'accuracy'] 1
'tableau'] 1
'adaptability' 1
'solve'] 1
'quantitative'] 1
'deep'] 1
'artificial'] 1
['analyze' 1
'evaluate'] 1
['communication'] 1
['accuracy' 1
['extract' 1
'predict' 1
['solve' 1
['data'] 1
['explore' 1
'explore'] 1
'analytical'] 1
'consolidate' 1
'collect' 1
['identify' 1
'collaboration'] 1
['algorithmic' 1
['design' 1
```

In [98]:
```python
#seeing the top keywords
word_counts.head()
```

Out[98]:
```
'data'              672
'communication'      69
'model'              67
'machine'            57
'analytical'         52
Name: job description, dtype: int64
```

In [99]:
```python
# Create a DataFrame with words and counts
word_counts_df = pd.DataFrame({'Word': words, 'Count': counts})

# Display the DataFrame
print(word_counts_df)
```

```
                Word  Count
0             'data'    672
1    'communication'     69
2            'model'     67
3          'machine'     57
4       'analytical'     52
..               ...    ...
94         'collect'      1
95       ['identify'      1
96  'collaboration']      1
97    ['algorithmic'      1
98         ['design'      1

[99 rows x 2 columns]
```

In [100…
```python
# Remove "]" and "[" from the dataset
word_counts_df['Word'] = word_counts_df['Word'].str.replace('[\[\]]', '', regex=True)

# Print the updated dataset
print(word_counts_df['Word'])
```

```
0              'data'
1     'communication'
2             'model'
3           'machine'
4        'analytical'
          ...
94          'collect'
95         'identify'
96    'collaboration'
97      'algorithmic'
98           'design'
Name: Word, Length: 99, dtype: object
```

In [101…
```python
#dropping the outlier word data
word_counts_df = word_counts_df[word_counts_df['Word'] != "'data'"]
word_counts_df
```

Out[101]:

	Word	Count
1	'communication'	69
2	'model'	67
3	'machine'	57
4	'analytical'	52
5	'technical'	48
...	...	...
94	'collect'	1
95	'identify'	1
96	'collaboration'	1
97	'algorithmic'	1
98	'design'	1

95 rows × 2 columns

In [102…
```python
# Prepare the data (assuming word_counts_df contains the words and counts)
X = word_counts_df[['Count']]  # Use the count values as the feature
X
```

Out[102]:

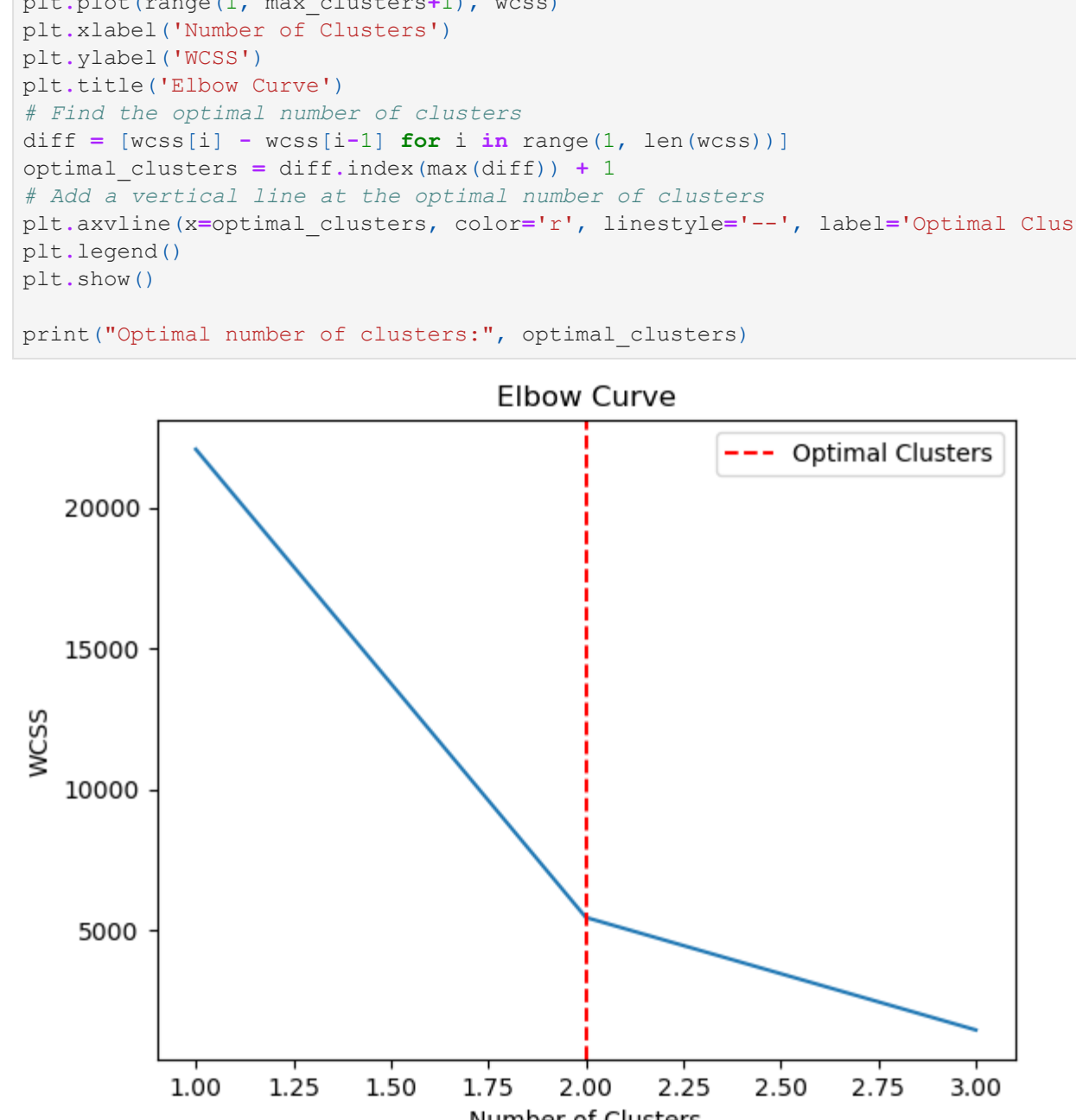
	Count
1	69
2	67
3	57
4	52
5	48
...	...
94	1
95	1
96	1
97	1
98	1

95 rows × 1 columns

In [103…
```python
# Set the range of clusters to try
max_clusters = 3

# Initialize an empty list to store the within-cluster sum of squares (WCSS)
wcss = []

# Fit K-Means clustering for different number of clusters
for n_clusters in range(1, max_clusters+1):
    kmeans = KMeans(n_clusters=n_clusters)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
```

```
C:\Users\sujoydutta\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1334: UserWarning: KMeans is known t
o have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it b
y setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\sujoydutta\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1334: UserWarning: KMeans is known t
o have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it b
y setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\sujoydutta\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1334: UserWarning: KMeans is known t
o have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it b
y setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

In [104…
```python
# Plot the Elbow Curve
plt.plot(range(1, max_clusters+1), wcss)
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.title('Elbow Curve')
# Find the optimal number of clusters
diff = [wcss[i] - wcss[i-1] for i in range(1, len(wcss))]
optimal_clusters = diff.index(max(diff)) + 1
# Add a vertical line at the optimal number of clusters
plt.axvline(x=optimal_clusters, color='r', linestyle='--', label='Optimal Clusters')
plt.legend()
plt.show()

print("Optimal number of clusters:", optimal_clusters)
```




```
Optimal number of clusters: 2
```

In [105…
```python
# Assuming you have already determined the optimal number of clusters
n_clusters = 2

# Fit K-Means clustering with the optimal number of clusters
kmeans = KMeans(n_clusters)
kmeans.fit(X)

# Calculate the Silhouette Coefficient
silhouette_avg = silhouette_score(X, kmeans.labels_)
print("Silhouette Coefficient:", silhouette_avg)

# Calculate the Calinski-Harabasz Index
ch_score = calinski_harabasz_score(X, kmeans.labels_)
print("Calinski-Harabasz Index:", ch_score)
```

```
C:\Users\sujoydutta\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1334: UserWarning: KMeans is known t
o have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it b
y setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
Silhouette Coefficient: 0.7937818373971774
Calinski-Harabasz Index: 284.12133465635605
```

In [106…
```python
# Perform clustering on the dataset using KMeans
kmeans = KMeans(n_clusters=2, random_state=42)
cluster_labels = kmeans.fit_predict(X)
```

```
C:\Users\sujoydutta\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1334: UserWarning: KMeans is known t
o have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it b
y setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

In [107…
```python
# Add the cluster labels to the dataset
word_counts_df['cluster_label'] = cluster_labels
```

In [108…
```python
#seeing the new dataset with labels
word_counts_df.head()
```

Out[108]:

	Word	Count	cluster_label
1	'communication'	69	1
2	'model'	67	1
3	'machine'	57	1
4	'analytical'	52	1
5	'technical'	48	1

In [113…
```python
# Create a word cloud for each cluster
for cluster in range(2):
    cluster_words = word_counts_df[word_counts_df['cluster_label']  == cluster]['Word']

    # Check if there are words available for the cluster
    if len(cluster_words) > 0:
        cluster_wordcloud = WordCloud().generate(' '.join(cluster_words))

        # Display the word cloud
        plt.figure(figsize=(8, 6))
        plt.imshow(cluster_wordcloud, interpolation='bilinear')
        plt.title('Cluster ' + str(cluster))
        plt.axis('off')
        plt.show()
    else:
        print(f"No words available for Cluster {cluster}")
```

Cluster 0

Cluster 1