1. Install and Import Libraries

I'll start by installing and importing the necessary libraries.

```python
# Import required libraries
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

# Allow inline plotting for visualizations in Jupyter
%matplotlib inline
```

2. Import and Clean the Data

I will load the Titanic dataset and clean it by removing any categorical variables. Since k-means works only with numerical data, I'll drop non-numeric columns such as Name, Sex, and Embarked. After that, I'll standardize the data to prevent bias from variables with different scales.

```python
# Define the path to your Titanic dataset
path = r'C:\Users\Asus\Music\achievement 6 project'

# Load the Titanic dataset
data = pd.read_csv(os.path.join(path, 'Data', 'tested.csv'), index_col=False)

# Display the first few rows of the dataset to understand its structure
data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

Cleaning and Processing the Data:

I'll drop categorical columns that can't be used for k-means.

I also check for missing values and decide whether to impute or drop them based on their significance.

```python
# Drop irrelevant and categorical columns
data_cleaned = data.drop(columns=['Name', 'Sex', 'Ticket', 'Cabin', 'Embarked'])

# Handling missing data (if necessary)
data_cleaned = data_cleaned.dropna()

# Display the cleaned data to verify
data_cleaned.head()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	892	0	3	34.5	0	0	7.8292
1	893	1	3	47.0	1	0	7.0000
2	894	0	2	62.0	0	0	9.6875
3	895	0	3	27.0	0	0	8.6625
4	896	1	3	22.0	1	1	12.2875

Standardizing the Data

```python
# Standardize the data so that all features are on a similar scale
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data_cleaned)

# Display the scaled data
scaled_data[:5]
```

```
array([[-1.69675037, -0.78901776,  1.01542612,  0.30665727, -0.55327231,
        -0.49211953, -0.54228095],
       [-1.68860203,  1.2673986 ,  1.01542612,  1.19423645,  0.59130978,
        -0.49211953, -0.55584416],
       [-1.68045369, -0.78901776, -0.16804587,  2.25933148, -0.55327231,
        -0.49211953, -0.51188479],
       [-1.67230535, -0.78901776,  1.01542612, -0.22589024, -0.55327231,
        -0.49211953, -0.52865069],
       [-1.66415701,  1.2673986 ,  1.01542612, -0.58092192,  0.59130978,
         0.74190748, -0.46935665]])
```

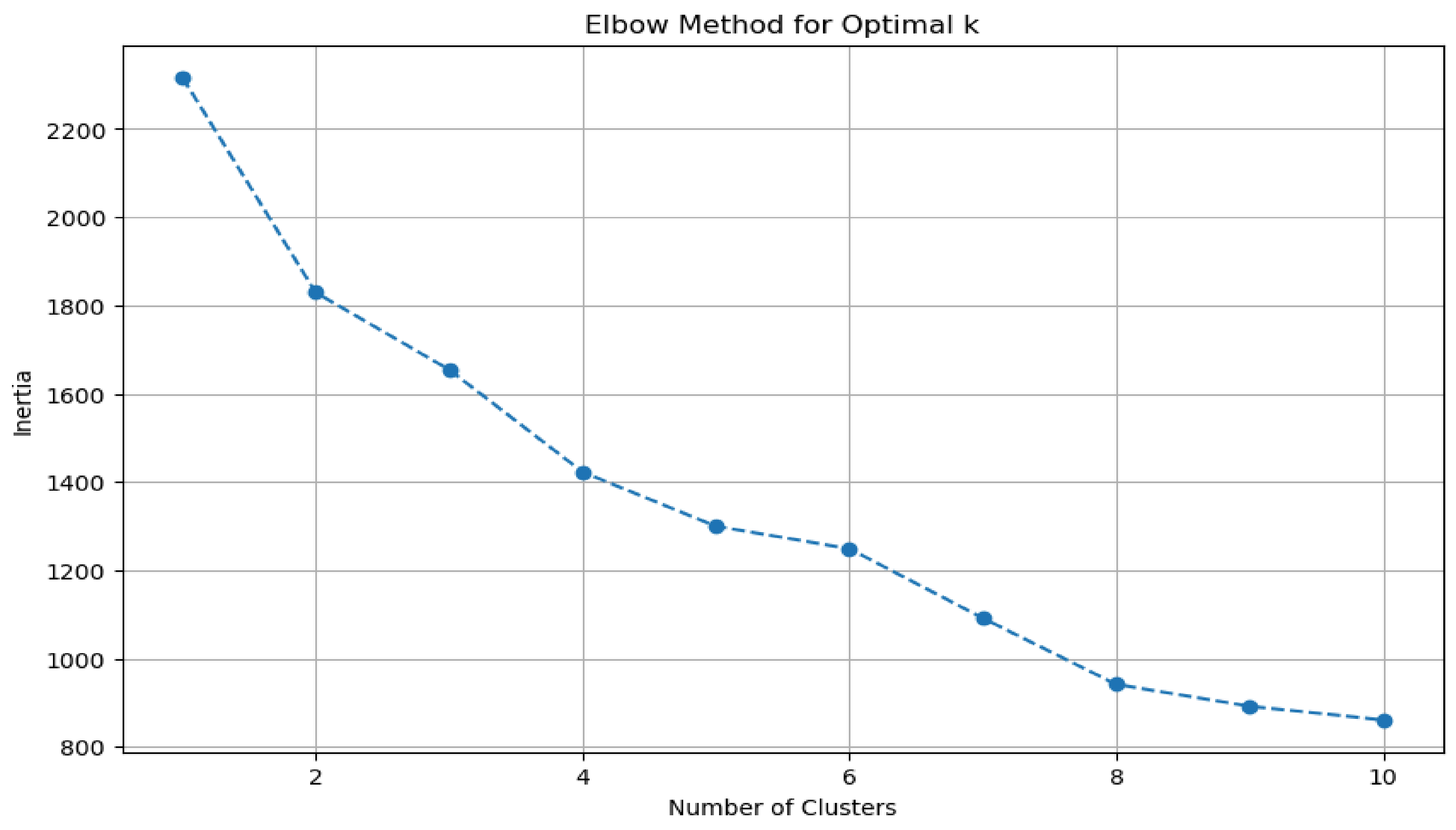
3. Use the Elbow Technique

I'll use the elbow technique to determine the optimal number of clusters by running k-means clustering for different numbers of clusters (from 1 to 10) and plotting the sum of squared distances (inertia) for each.

```python
# Initialize list to store inertia values
inertia = []

# Compute inertia for cluster numbers from 1 to 10
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_data)
    inertia.append(kmeans.inertia_)

# Plot the elbow chart
plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), inertia, marker='o', linestyle='--')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.grid(True)
plt.show()
```



4. Choose the Number of Clusters

Rationale:

The elbow point is where the rate of decrease in inertia significantly slows down, indicating diminishing returns on increasing the number of clusters. In the provided chart, the inertia drops sharply as the number of clusters increases from 2 to 4. After 4 clusters, the decrease in inertia becomes more gradual and less steep. Choosing 4 clusters balances the complexity of the model (in terms of the number of clusters) and the compactness of the clusters (as indicated by lower inertia).

Potential Issues:

If the elbow point is not very clear or if different runs of the algorithm suggest different elbow points, it might be challenging to decisively determine the optimal number of clusters. In such cases, additional methods like silhouette analysis or consulting domain expertise could provide further insights. This approach ensures that the k-means clustering algorithm is likely to perform effectively without overfitting by having too many clusters, or underfitting by having too few.

Based on the elbow chart analysis, we've determined that 4 clusters is the optimal number for the k-means algorithm. This decision balances model complexity and cluster compactness.

5. Run the k-means Algorithm

```python
# Initialize k-means with the optimal number of clusters
kmeans = KMeans(n_clusters=4, random_state=42)  # Replace X with the number of clusters

# Fit k-means and predict cluster labels
kmeans.fit(scaled_data)
data_cleaned['Cluster'] = kmeans.labels_

# Display the dataset with cluster labels
data_cleaned.head()
```

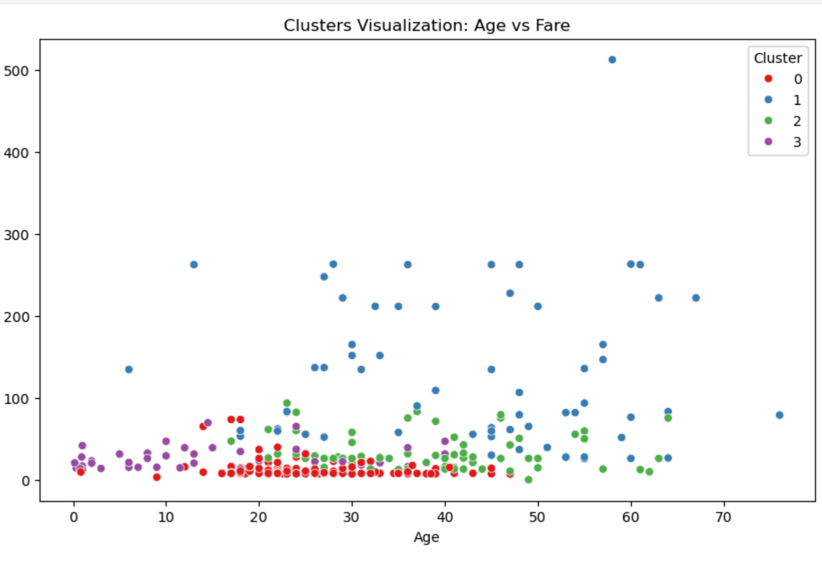
	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	Cluster
0	892	0	3	34.5	0	0	7.8292	0
1	893	1	3	47.0	1	0	7.0000	0
2	894	0	2	62.0	0	0	9.6875	2
3	895	0	3	27.0	0	0	8.6625	0
4	896	1	3	22.0	1	1	12.2875	3

6. Create Visualizations

6.1 Scatterplot of Age vs Fare

Create a scatter plot to visualize how different clusters are distributed across age and fare.

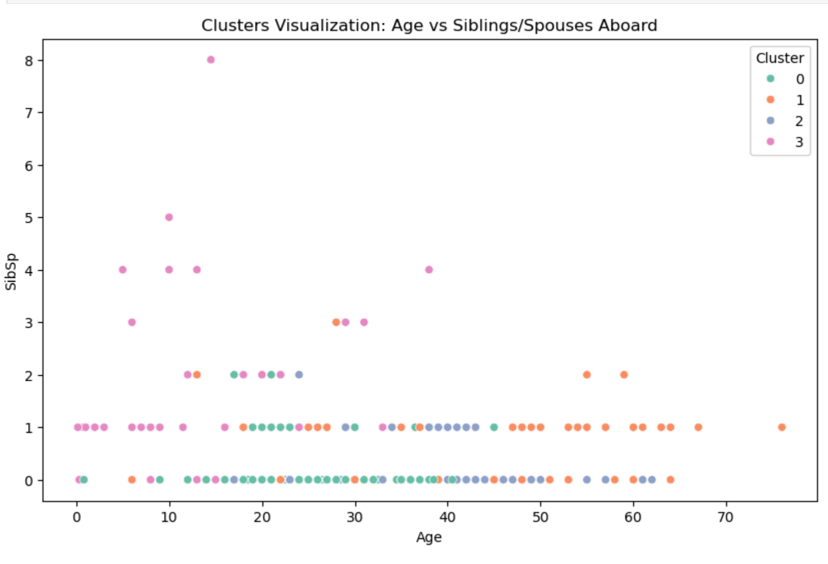
```python
# Scatterplot: Age vs Fare colored by cluster
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Age', y='Fare', hue='Cluster', data=data_cleaned, palette='Set1')
plt.title('Clusters Visualization: Age vs Fare')
plt.xlabel('Age')
plt.ylabel('Fare')
plt.legend(title='Cluster')
plt.show()
```



7 Additional Scatterplot: Age vs SibSp

Create an additional scatter plot to further explore cluster distribution.

```python
# Scatterplot: Age vs SibSp (Siblings/Spouses Aboard) colored by cluster
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Age', y='SibSp', hue='Cluster', data=data_cleaned, palette='Set2')
plt.title('Clusters Visualization: Age vs Siblings/Spouses Aboard')
plt.xlabel('Age')
plt.ylabel('SibSp')
plt.legend(title='Cluster')
plt.show()
```



8. Discuss the Clusters

Cluster 0 (Red):

Concentration: Lower fare range, mostly below 100, across all age groups. Interpretation: This cluster likely represents economy or standard fare passengers. The wide age range suggests it includes individuals from various demographics, possibly including budget-conscious travelers, families, and seniors traveling on lower-cost tickets. Significance: This group forms a substantial portion of the passenger base, indicating a strong market for lower-priced fares across all ages.

Cluster 1 (Blue):

Concentration: Widest spread, covering all age groups and fare ranges, including the highest fares. Interpretation: This cluster appears to represent a diverse group of passengers, possibly including premium or first-class travelers, as well as those who booked last-minute or during peak periods. The wide spread suggests variability in purchasing behavior or ticket types. Significance: The presence of high-fare outliers in this cluster, especially the notable one around age 60 with a fare of about 500, indicates a market for luxury or premium services.

Cluster 2 (Green):

Concentration: Middle fare range, mostly between 50 and 100, across various age groups. Interpretation: This cluster might represent mid-tier ticket holders, possibly including business travelers or those opting for slightly upgraded services compared to the basic fare. Significance: The existence of this distinct middle cluster suggests a market segment that values some additional comforts or services but is not opting for the highest-priced options.

Cluster 3 (Purple):

Concentration: Younger age groups (roughly 0-30 years) and lower fare ranges. Interpretation: This cluster likely represents younger travelers, possibly including students, young professionals, or families with young children. The concentration in lower fare ranges suggests price sensitivity among this group. Significance: The distinct clustering of younger ages indicates a potential for age-specific marketing or services tailored to younger travelers.

Key Observations and Implications:

Age-Fare Relationship: While there isn't a strong linear correlation between age and fare, the clustering reveals some age-related patterns, particularly for younger travelers (Cluster 3).

Price Segmentation: The clear separation into different fare ranges (low, middle, high) across clusters suggests effective price segmentation strategies.

Diverse Customer Base: The spread across all age groups in multiple clusters indicates a diverse customer base with varying willingness or ability to pay.

Premium Market: The existence of high-fare data points, especially in Cluster 1, suggests a viable market for premium or luxury services.

Youth Market: The distinct cluster for younger ages (Cluster 3) in lower fare ranges points to a significant youth or student market that might benefit from targeted promotions or services.

Potential for Targeted Marketing: The clear clustering provides opportunities for targeted marketing strategies, tailoring offerings to different age groups and price sensitivities.

Pricing Strategy Insights: The distribution of clusters can inform pricing strategies, helping to optimize fare structures for different customer segments.

This cluster analysis provides valuable insights into customer segmentation, pricing strategies, and potential marketing approaches. It suggests a need for diverse service offerings to cater to different age groups and price points, while also highlighting opportunities for upselling or creating targeted packages for specific clusters.

9. Calculate Descriptive Statistics

Calculate descriptive statistics for each cluster to understand their characteristics.

```python
# Calculate mean statistics for each cluster
cluster_stats = data_cleaned.groupby('Cluster').mean()

# Display the descriptive statistics
cluster_stats
```

Cluster	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	1099.436242	0.322148	2.744966	25.045839	0.140940	0.046980	12.015688
1	1115.825397	0.761905	1.063492	42.801587	0.682540	0.714286	130.683865
2	1081.109589	0.041096	1.479452	38.609589	0.356164	0.082192	32.892522
3	1111.804348	0.608696	2.717391	16.155870	1.521739	1.608696	24.793387

Here's a detailed explanation of the findings for each cluster:

Cluster 0

PassengerId: The average PassengerId is approximately 1099.43. Survived: About 32.21% of the passengers in this cluster survived. Pclass: The passengers predominantly belong to a lower class (average of 2.74). Age: The average age in this cluster is around 25.05 years. SibSp: Passengers in this cluster have an average of 0.14 siblings or spouses aboard. Parch: They have an average of 0.05 parents or children aboard. Fare: The average fare paid by passengers in this cluster is approximately 12.02.

Cluster 1

PassengerId: The average PassengerId is approximately 1115.82. Survived: A higher survival rate is observed here, with about 76.19% of the passengers surviving. Pclass: This cluster has passengers primarily from a higher class (average of 1.06). Age: The average age is significantly higher at about 42.80 years. SibSp: There is a higher average of siblings or spouses aboard (0.68). Parch: Also, a higher average of parents or children aboard (0.71). Fare: Passengers in this cluster paid a much higher average fare of approximately 130.68.

Cluster 2

PassengerId: The average PassengerId is approximately 1081.11. Survived: This cluster has a very low survival rate, with only about 4.11% surviving. Pclass: The average class is around 1.47, indicating a mix but slightly better than middle class. Age: The average age here is about 38.61 years. SibSp: An average of 0.35 siblings or spouses are aboard. Parch: A low average of 0.08 parents or children aboard. Fare: The average fare for this cluster is approximately

Cluster 3

PassengerId: The average PassengerId is approximately 1111.80. Survived: About 60.87% of the passengers in this cluster survived. Pclass: The passengers are from a slightly higher class on average (2.17). Age: The average age is lower at about 16.16 years. SibSp: There is a high average of siblings or spouses aboard (1.52). Parch: Similarly, a high average of parents or children aboard (1.61). Fare: The average fare paid is approximately

Summary

Cluster 0 represents younger, lower-class passengers with minimal family aboard and lower fare costs. Cluster 1 includes older, higher-class passengers with more family members aboard and significantly higher fare costs, showing the highest survival rate. Cluster 2 seems to consist of middle-aged passengers with very few family members aboard, moderate fare costs, and the lowest survival rate. Cluster 3 includes very young passengers, likely children, with many family members aboard, paying moderate fares and having a relatively high survival rate. 24.79. 32.89.

10. Propose Future Steps

Here are the proposed future steps based on the analysis of the clusters:

In-depth Statistical Analysis:

Conduct a multivariate analysis to identify significant predictors of survival. This could include logistic regression or decision tree analysis to understand the impact of variables like age, class, and family size on survival rates.

Data Visualization:

Create visualizations such as heatmaps, bar charts, or scatter plots to better understand the relationships between variables. This can help in identifying patterns or anomalies that are not immediately apparent in the raw data.

Feature Engineering:

Consider creating new features that might capture more complex relationships, such as interaction terms between class and fare or age and family size. This could improve the predictive power of any models developed.

Model Development:

Develop predictive models using machine learning techniques such as Random Forest, Gradient Boosting, or Neural Networks. These models can be used to predict survival based on the identified features.

Cross-validation and Model Tuning:

Implement cross-validation techniques to ensure the robustness of the models. Fine-tune the model parameters to achieve the best performance.

Cluster Analysis Refinement:

Re-evaluate the clustering algorithm and parameters used. Consider using different clustering techniques like K-means, hierarchical clustering, or DBSCAN to see if more meaningful clusters can be identifiedExploration of Additional Data:

If available, incorporate additional data sources that might provide more context or variables that could influence survival, such as socio-economic status or health conditions.

Ethical and Social Implications:

Consider the ethical implications of the analysis, especially if the data is used for decision-making. Ensure that the analysis does not reinforce biases or lead to unfair treatment of certain groups.

Reporting and Communication:

Prepare a comprehensive report detailing the findings, methodologies, and implications of the analysis. Use clear and concise language to communicate the results to stakeholders who may not have a technical background.