In [79]:

```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()  # for plot styling
import numpy as np
import pandas as pd
```

In [105]:

```
df = pd.read_csv(r"C:\Users\mdmoh\Downloads\train.csv")
df = df.dropna()
df.head()
df.dtypes
df.astype({'Fare': 'int64', 'Age':'int64'}).dtypes
```

Out[105]:

```
PassengerId     int64
Survived        int64
Pclass          int64
Name           object
Sex            object
Age             int64
SibSp           int64
Parch           int64
Ticket         object
Fare            int64
Cabin          object
Embarked       object
dtype: object
```

In [106]:

```
X=df.iloc[:,[5,9]].values
print(X)
```

```
[[ 38.      71.2833]
 [ 35.      53.1   ]
 [ 54.      51.8625]
 [  4.      16.7   ]
 [ 58.      26.55  ]
 [ 34.      13.    ]
 [ 28.      35.5   ]
 [ 19.     263.    ]
 [ 49.      76.7292]
 [ 65.      61.9792]
 [ 45.      83.475 ]
 [ 29.      10.5   ]
 [ 25.       7.65  ]
 [ 23.     263.    ]
 [ 46.      61.175 ]
 [ 71.      34.6542]
 [ 23.      63.3583]
 [ 21.      77.2875]
 [ 47.      52.    ]
 [ 24.     247.5208]
 [ 32.5     13.    ]
 [ 54.      77.2875]
 [ 19.      26.2833]
 [ 37.      53.1   ]
 [ 24.      79.2   ]
 [ 36.5     26.    ]
 [ 22.      66.6   ]
 [ 61.      33.5   ]
 [ 56.      30.6958]
 [ 50.      28.7125]
 [  1.      39.    ]
 [  3.      26.    ]
 [ 44.      27.7208]
 [ 58.     146.5208]
 [  2.      10.4625]
 [ 40.      31.    ]
 [ 31.     113.275 ]
 [ 32.      76.2917]
 [ 38.      90.    ]
 [ 35.      83.475 ]
 [ 44.      90.    ]
 [ 37.      52.5542]
 [ 29.      10.4625]
 [ 62.      26.55  ]
 [ 30.      86.5   ]
```

```
[ 52.      79.65  ]
[ 40.       0.    ]
[ 58.     153.4625]
[ 35.     135.6333]
[ 37.      29.7   ]
[ 63.      77.9583]
[ 19.      91.0792]
[ 36.      12.875 ]
[  2.     151.55  ]
[ 50.     247.5208]
[  0.92   151.55  ]
[ 17.     108.9   ]
[ 30.      56.9292]
[ 24.      83.1583]
[ 18.     262.375 ]
[ 31.     164.8667]
[ 40.     134.5   ]
[ 36.     135.6333]
[ 36.      13.    ]
[ 16.      57.9792]
[ 45.5     28.5   ]
[ 38.     153.4625]
[ 29.      66.6   ]
[ 41.     134.5   ]
[ 45.      35.5   ]
[  2.      26.    ]
[ 24.     263.    ]
[ 24.      13.    ]
[ 22.      55.    ]
[ 60.      75.25  ]
[ 24.      69.3   ]
[ 25.      55.4417]
[ 27.     211.5   ]
[ 36.     120.    ]
[ 23.     113.275 ]
[ 24.      16.7   ]
[ 33.      90.    ]
[ 32.       8.05  ]
[ 28.      26.55  ]
[ 50.      55.9   ]
[ 14.     120.    ]
[ 64.     263.    ]
[  4.      81.8583]
[ 52.      30.5   ]
[ 30.      27.75  ]
[ 49.      89.1042]
[ 65.      26.55  ]
[ 48.      26.55  ]
[ 47.      38.5   ]
[ 23.      13.7917]
[ 25.      91.0792]
[ 35.      90.    ]
[ 58.      29.7   ]
[ 55.      30.5   ]
[ 54.      78.2667]
[ 25.     151.55  ]
[ 16.      86.5   ]
[ 18.     108.9   ]
[ 36.      26.2875]
[ 47.      34.0208]
[ 34.      10.5   ]
[ 30.      93.5   ]
[ 44.      57.9792]
[ 45.      26.55  ]
[ 22.      49.5   ]
[ 36.      71.    ]
[ 50.     106.425 ]
[ 17.     110.8833]
[ 48.      39.6   ]
[ 39.      79.65  ]
[ 53.      51.4792]
[ 36.      26.3875]
[ 39.      55.9   ]
[ 39.     110.8833]
[ 36.      40.125 ]
[ 18.      79.65  ]
[ 60.      79.2   ]
[ 52.      78.2667]
[ 49.      56.9292]
[ 40.     153.4625]
[  4.      39.    ]
[ 42.      52.5542]
[ 61.      32.3208]
```

```
 [ 21.      77.9583]
 [ 80.      30.    ]
 [ 32.      30.5   ]
 [ 24.      69.3   ]
 [ 48.      76.7292]
 [ 56.      35.5   ]
 [ 58.     113.275 ]
 [ 47.      25.5875]
 [ 31.      52.    ]
 [ 36.     512.3292]
 [ 27.      76.7292]
 [ 15.     211.3375]
 [ 31.      57.    ]
 [ 49.     110.8833]
 [ 42.       7.65  ]
 [ 18.     227.525 ]
 [ 35.      26.2875]
 [ 42.      26.2875]
 [ 24.      49.5042]
 [ 48.      52.    ]
 [ 19.       7.65  ]
 [ 38.     227.525 ]
 [ 27.      10.5   ]
 [ 27.      53.1   ]
 [ 29.     211.3375]
 [ 35.     512.3292]
 [ 36.      78.85  ]
 [ 21.     262.375 ]
 [ 70.      71.    ]
 [ 19.      53.1   ]
 [  6.      12.475 ]
 [ 33.      86.5   ]
 [ 36.     120.    ]
 [ 51.      77.9583]
 [ 57.      10.5   ]
 [ 43.     211.3375]
 [ 17.      57.    ]
 [ 29.      30.    ]
 [ 46.      79.2   ]
 [ 49.      25.9292]
 [ 11.     120.    ]
 [ 39.       0.    ]
 [ 33.      53.1   ]
 [ 52.      93.5   ]
 [ 27.      12.475 ]
 [ 39.      83.1583]
 [ 16.      39.4   ]
 [ 51.      26.55  ]
 [ 48.      25.9292]
 [ 31.      50.4958]
 [ 47.      52.5542]
 [ 33.       5.    ]
 [ 56.      83.1583]
 [ 19.      30.    ]
 [ 26.      30.    ]]
```

In [107]:

```
y=df.iloc[:,9].values
y= y.reshape(-1,1)
```

In [108]:

```
print(type(X))
```

```
<class 'numpy.ndarray'>
```

In [109]:

```
print(X.shape, y.shape)
```

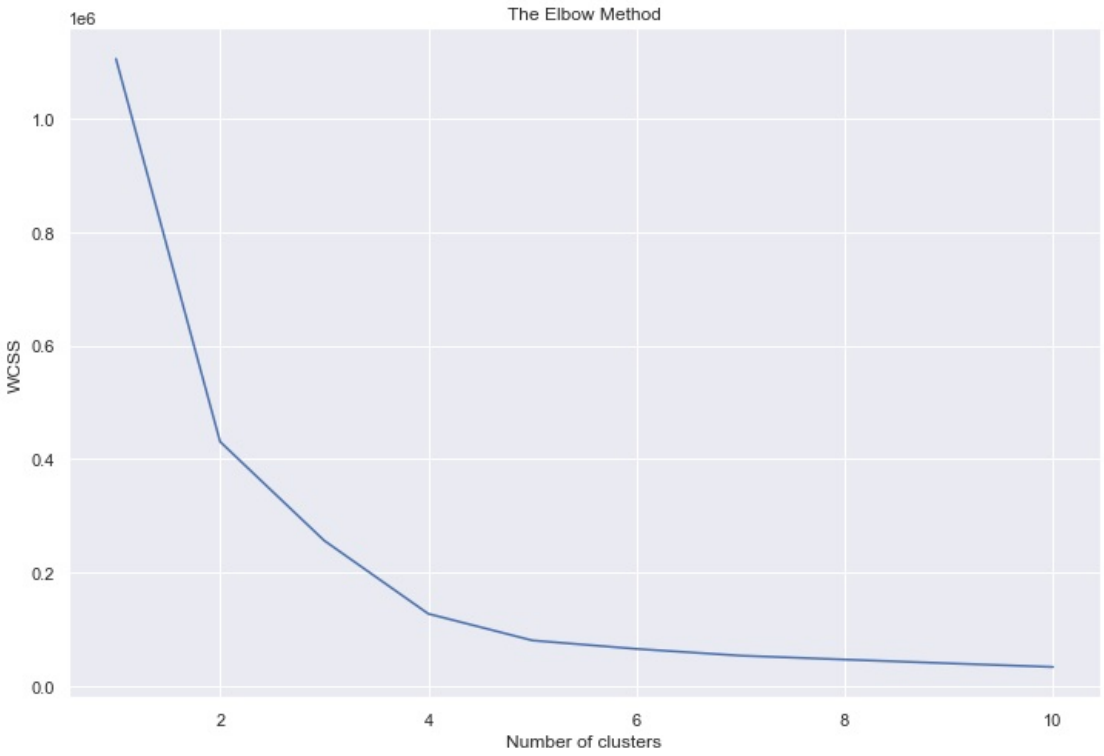
```
(183, 2) (183, 1)
```

In [110]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
y_train = sc_y.fit_transform(y_train.reshape(-1, 1))
X_train = sc_X.fit_transform(X_train.reshape(-1, 1))
X_test = sc_X.fit_transform(X_test.reshape(-1, 1))

sc_X = StandardScaler()
sc_y = StandardScaler()
```

In [111]:

```
from sklearn.cluster import KMeans

wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(12,8))
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



In [112]:

```
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 2)
y_kmeans = kmeans.fit_predict(X)
y_kmeans
```

Out[112]:

```
array([4, 4, 4, 1, 1, 1, 1, 2, 4, 4, 4, 1, 1, 2, 4, 1, 4, 4, 4, 2, 1, 4,
       1, 4, 4, 1, 4, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 4, 4, 4, 4, 4, 1, 1,
       4, 4, 1, 0, 0, 1, 4, 4, 1, 0, 2, 0, 0, 4, 4, 2, 0, 0, 0, 1, 4, 1,
       0, 4, 0, 1, 1, 2, 1, 4, 4, 4, 4, 2, 0, 0, 1, 4, 1, 1, 4, 0, 2, 4,
       1, 1, 4, 1, 1, 1, 1, 4, 4, 1, 1, 4, 0, 4, 0, 1, 1, 1, 4, 4, 1, 4,
       4, 0, 0, 1, 4, 4, 1, 4, 0, 1, 4, 4, 4, 4, 0, 1, 4, 1, 4, 1, 1, 4,
       4, 1, 0, 1, 4, 3, 4, 2, 4, 0, 1, 2, 1, 1, 4, 4, 1, 2, 1, 4, 2, 3,
       4, 2, 4, 4, 1, 4, 0, 4, 1, 2, 4, 1, 4, 1, 0, 1, 4, 4, 1, 4, 1, 1,
       1, 4, 4, 1, 4, 1, 1])
```

In [113]:

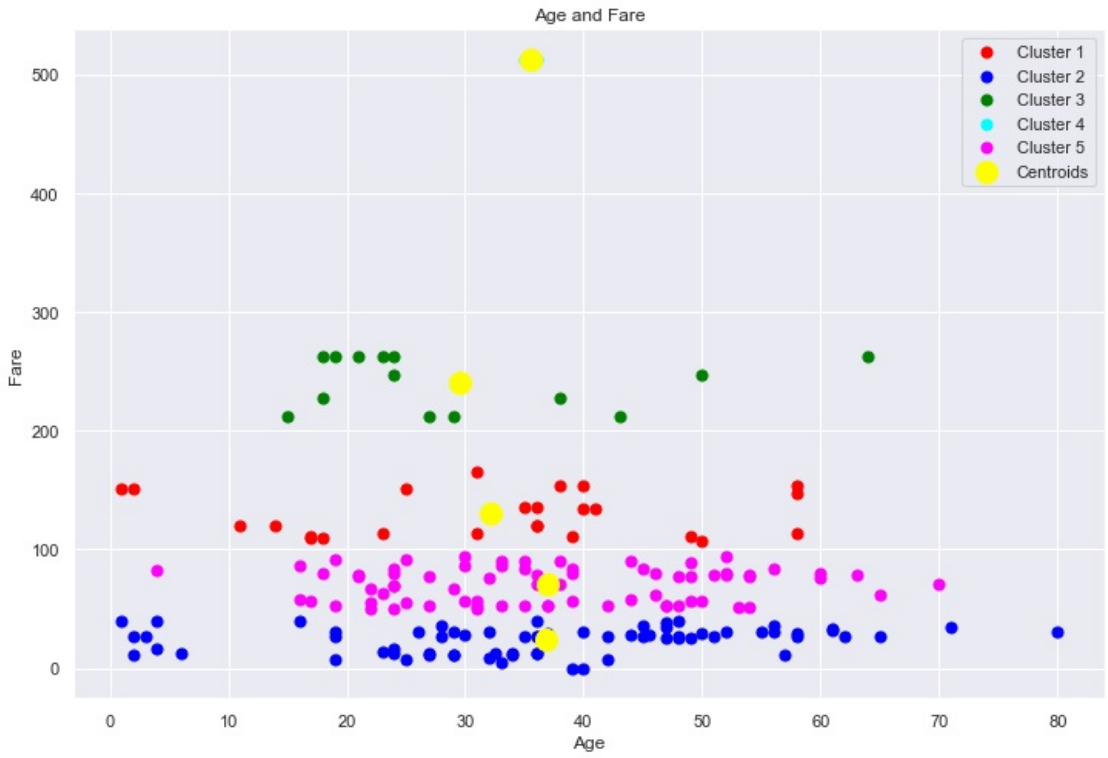
```
centroids = kmeans.cluster_centers_
centroids
```

Out[113]:

```
array([[ 32.1568    , 129.73566   ],
       [ 36.80434783,  23.63291014],
       [ 29.5       , 240.88243571],
       [ 35.5       , 512.3292    ],
       [ 37.        ,  70.24406849]])
```

In [115]:

```
plt.figure(figsize=(12,8))
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 50, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 50, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 50, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 50, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 50, c = 'magenta', label = 'Cluster 5')
plt.scatter(kmeans.cluster_centers_[:5, 0], kmeans.cluster_centers_[:5, 1], s = 200, c = 'yellow', label = 'Centr
oids')
plt.title('Age and Fare')
plt.xlabel('Age')
plt.ylabel('Fare')
plt.legend()
plt.show()
```



In [127]:

```
X[y_kmeans == 0]
```

Out[127]:

```
array([[ 58.    , 146.5208],
       [ 31.    , 113.275 ],
       [ 58.    , 153.4625],
       [ 35.    , 135.6333],
       [  2.    , 151.55  ],
       [  0.92  , 151.55  ],
       [ 17.    , 108.9   ],
       [ 31.    , 164.8667],
       [ 40.    , 134.5   ],
       [ 36.    , 135.6333],
       [ 38.    , 153.4625],
       [ 41.    , 134.5   ],
       [ 36.    , 120.    ],
       [ 23.    , 113.275 ],
       [ 14.    , 120.    ],
       [ 25.    , 151.55  ],
       [ 18.    , 108.9   ],
       [ 50.    , 106.425 ],
       [ 17.    , 110.8833],
       [ 39.    , 110.8833],
       [ 40.    , 153.4625],
       [ 58.    , 113.275 ],
       [ 49.    , 110.8833],
       [ 36.    , 120.    ],
       [ 11.    , 120.    ]])
```

In [123]:

```
X_c_1 = X[y_kmeans == 1]
X_c_1 = pd.DataFrame(X_c_1, columns=['Age','Fare'])
X_c_1
```

Out[123]:

	Age	Fare
0	4.0	16.7000
1	58.0	26.5500
2	34.0	13.0000
3	28.0	35.5000
4	29.0	10.5000
...	...	...
64	51.0	26.5500
65	48.0	25.9292
66	33.0	5.0000
67	19.0	30.0000
68	26.0	30.0000

69 rows × 2 columns

In [124]:

```
b=[5,15]
b
```

Out[124]:

```
[5, 15]
```

In [126]:

```

```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-126-8979d129e554> in <module>
----> 1 b[0,1]

TypeError: list indices must be integers or slices, not tuple
```

In [ ]: