In [ ]:

In [15]:
```python
import numpy as np # linear algebra
import matplotlib.pyplot as plt # For plotting
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

In [16]:
```python
# Importing the dataset
dataset = pd.read_csv(r"C:\Users\Anisha\Downloads\Social_Network_Ads.csv")
X = dataset.iloc[:,].values
y = dataset.iloc[:, -1].values
```

In [17]:
```python
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_s
```

In [18]:
```python
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [19]:
```python
# Fitting classifier to the Training set
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2)
classifier.fit(X_train, y_train)
```

Out[19]:
```
KNeighborsClassifier()
```

In [20]:
```python
# Predicting the Test set results
y_pred = classifier.predict(X_test)
```
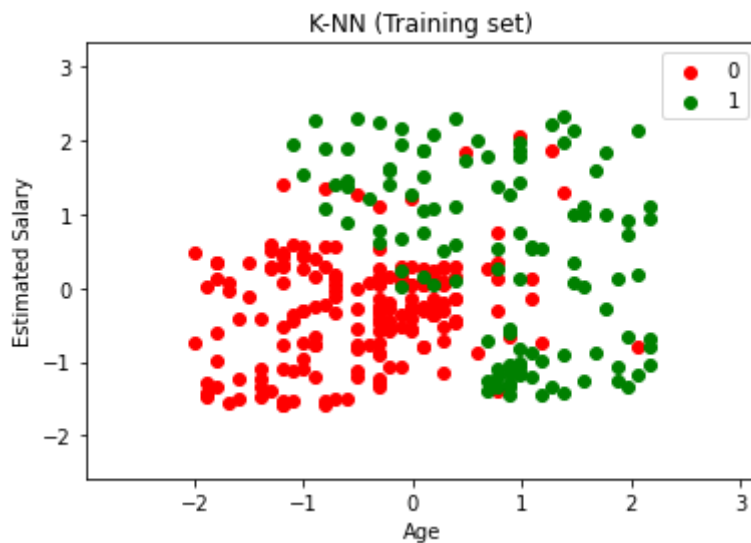
In [24]:
```python
# Visualising the Training set results
from matplotlib.colors import ListedColormap
X_set, y_set = X_train, y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max
#plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).resha
            #alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('K-NN (Training set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

c argument looks like a single numeric RGB or RGBA sequence, which should be avoid
ed as value-mapping will have precedence in case its length matches with *x* & *y*.
Please use the *color* keyword-argument or provide a 2D array with a single row if y

ou intend to specify the same RGB or RGBA value for all points.
c argument looks like a single numeric RGB or RGBA sequence, which should be avoid
ed as value-mapping will have precedence in case its length matches with *x* & *y*.
Please use the *color* keyword-argument or provide a 2D array with a single row if y
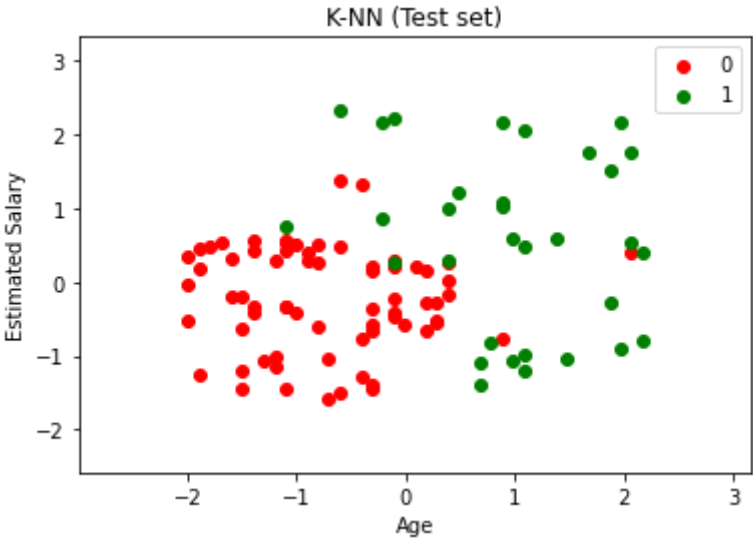ou intend to specify the same RGB or RGBA value for all points.

In [23]:
```python
# Visualising the Test set results
from matplotlib.colors import ListedColormap
X_set, y_set = X_test, y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max
#plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).resha
            #alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('K-NN (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

c argument looks like a single numeric RGB or RGBA sequence, which should be avoid
ed as value-mapping will have precedence in case its length matches with *x* & *y*.
Please use the *color* keyword-argument or provide a 2D array with a single row if y
ou intend to specify the same RGB or RGBA value for all points.
c argument looks like a single numeric RGB or RGBA sequence, which should be avoid
ed as value-mapping will have precedence in case its length matches with *x* & *y*.
Please use the *color* keyword-argument or provide a 2D array with a single row if y
ou intend to specify the same RGB or RGBA value for all points.

In [ ]:

In [ ]:

In [ ]: