```
import numpy as Sidharth_np
import matplotlib.pyplot as Sidharth_plt
import pandas as Sidharth_pd
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
5	15728773	Male	27	58000	0
6	15598044	Female	27	84000	0
7	15694829	Female	32	150000	1
8	15600575	Male	25	33000	0
9	15727311	Female	35	65000	0

Next steps: View recommended plots

Importing dataset

```
ds = Sidharth_pd.read_csv("/content/sample_data/KNN_Data1.csv")
ds.head(10)
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
5	15728773	Male	27	58000	0
6	15598044	Female	27	84000	0
7	15694829	Female	32	150000	1
8	15600575	Male	25	33000	0
9	15727311	Female	35	65000	0

Next steps: View recommended plots

Extracting Independent anddependent Variables

```
x = ds.iloc[:,[2,3]].values
y = ds.iloc[:,4].values
```

Splitting datasetinto training 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_state=0)
```

Feature scaling

```
from sklearn.preprocessing import StandardScaler
st_x=StandardScaler()
x_train=st_x.fit_transform(x_train)
x_test=st_x.transform(x_test)
```

FittingK-NN 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)
KNeighborsClassifier()
```

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

Predicting the test set result

```
y_pred = classifier.predict(x_test)
y_pred
```

```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1])
```

Creating confusionmatrix

```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test, y_pred)
cm
```

```
array([[64,  4],
       [ 3, 29]])
```

Visualising the trianing set result

```
from matplotlib.colors import ListedColormap
x_set, y_set = x_train, y_train
x1, x2 =Sidharth_np.meshgrid(Sidharth_np.arange(start = x_set[:,
0].min() - 1, stop = x_set[:, 0].max() + 1, step =0.01),
Sidharth_np.arange(start = x_set[:, 1].min() - 1, stop = x_set[:,
1].max() + 1, step = 0.01))
Sidharth_plt.contourf(x1, x2,
classifier.predict(Sidharth_np.array([x1.ravel(),
x2.ravel()]).T).reshape(x1. shape), alpha = 0.75, cmap =
ListedColormap(('black','black' )))
Sidharth_plt.xlim(x1.min(), x1.max())
Sidharth_plt.ylim(x2.min(), x2.max())
for i, j in enumerate(Sidharth_np.unique(y_set)):
  Sidharth_plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1], c= ListedColormap(('orange', 'white'))(i), label = j)
  Sidharth_plt.title('K-NN Algorithm (Training set)')
  Sidharth_plt.xlabel('Age')
  Sidharth_plt.ylabel('Estimated Salary')
Sidharth_plt.legend()
Sidharth_plt.show()
```

```
<ipython-input-22-d120a03be4b5>:14: UserWarning: *c* argument looks like a single nur
  Sidharth_plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1], c= ListedColormap
```

Visualising the test result

```
from matplotlib.colors import ListedColormap
x_set, y_set = x_test, y_test
x1, x2 = Sidharth_np.meshgrid(Sidharth_np.arange(start = x_set[:,
0].min() - 1, stop = x_set[:, 0].max() + 1, step=0.01),
Sidharth_np.arange(start = x_set[:, 1].min() - 1, stop = x_set[:,
1].max() + 1, step = 0.01))
Sidharth_plt.contourf(x1, x2,
classifier.predict(Sidharth_np.array([x1.ravel(),
x2.ravel()]).T).reshape(x1. shape), alpha = 0.75, cmap =
ListedColormap(('black', 'black')))
Sidharth_plt.xlim(x1.min(), x1.max())
Sidharth_plt.ylim(x2.min(), x2.max())
for i, j in enumerate(Sidharth_np.unique(y_set)):
  Sidharth_plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1], c= ListedColormap(('orange','white'))(i), label = j)
  Sidharth_plt.title('K-NN algorithm(Test set)')
  Sidharth_plt.xlabel('Age')
  Sidharth_plt.ylabel('Estimated Salary')
Sidharth_plt.legend()
Sidharth_plt.show()
```

```
<ipython-input-24-41d0e11a7d88>:14: UserWarning: *c* argument looks like a single nur
  Sidharth_plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1], c= ListedColormap
```