knn2

March 1, 2021

[87]:

```
import numpy as np
from collections import Counter


def euclidean_distance(x1, x2):
        return np.sqrt(np.sum((x1 - x2)**2))


class K:

    def __init__(self, k=3):
        self.k = k

    def fit(self, X, y):
        self.X_train = X
        self.y_train = y

    def predict(self, X):
        y_pred = [self._predict(x) for x in X]
        return np.array(y_pred)

    def _predict(self, x):
        distances = [euclidean_distance(x, x_train) for x_train in self.X_train]
        k_idx = np.argsort(distances)[:self.k]
        k_neighbor_labels = [self.y_train[i] for i in k_idx]
        most_common = Counter(k_neighbor_labels).most_common(1)
        return most_common[0][0]
```

[93]:

```
import numpy as np
import pandas as pd
from sklearn import datasets
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap


cm = ListedColormap(['#FF0000', '#0037ff', '#00ffb3'])
```

```
def accuracy(y_true, y_pred):
    acc = np.sum(y_true == y_pred) / len(y_true)
    return acc



X  = np.random.normal(loc=-2.0, scale=2.0, size=int(n/2))
y = np.random.normal(loc=0, scale=1, size=int(n/2))
X, y = X, y


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=4)




k = 3
clf = KNN(k=k)
clf.fit(X_train, y_train)
predictions = clf.predict(X_test)
print("accuracy = ", accuracy(y_test, predictions))
predictions = clf.predict(X_test)
```

```
accuracy =  0.0
```

[ ]:

```
```

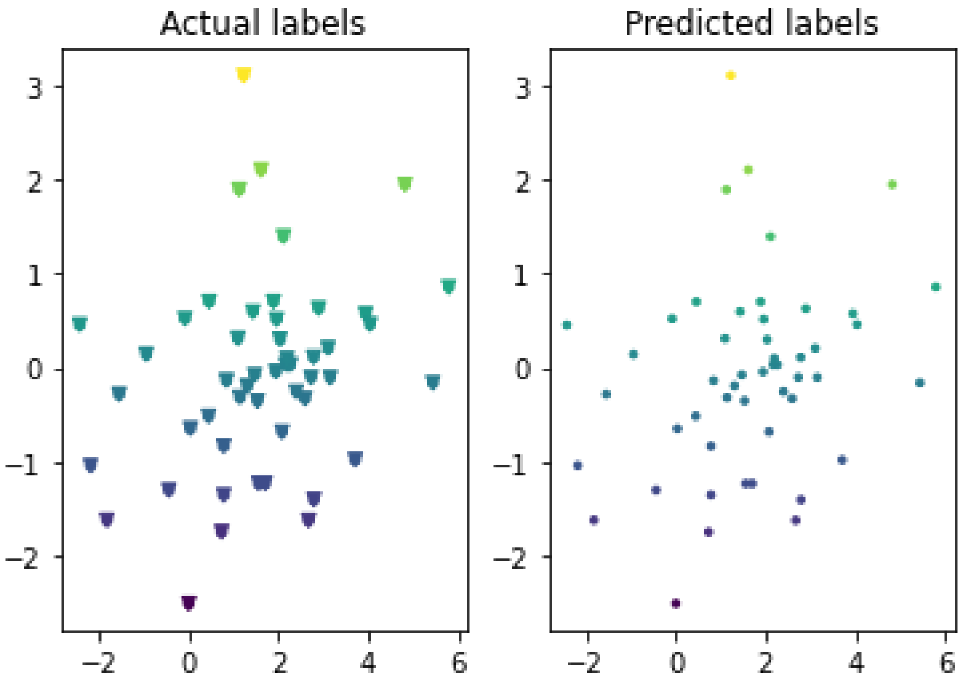
[91]:

```
X1 = np.random.normal(loc=2.0, scale=2.0, size=int(n/2))
X2 = np.random.normal(loc=0, scale=1, size=int(n/2))
X, y = X1, X2
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=4)



plt.subplot(1, 2, 1)
plt.scatter(X,y, s=50, c=y, marker=".")
plt.scatter(X,y, s=25, c=y, marker="v")
plt.title("Actual labels")

plt.subplot(1, 2, 2)
plt.title("Predicted labels")
plt.scatter(X,y, s=25, c=y, marker=".")
plt.scatter(X,y, s=20, c=predictions, marker="v")
plt.title("Predicted labels")
plt.tight_layout()
plt.show()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~/Desktop/HomeWork/SJSU/cs156/anaconda/lib/python3.8/site-packages/matplotlib/
 ↪axes/_axes.py in _parse_scatter_color_args(c, edgecolors, kwargs, xsize,␣
 ↪get_next_color_func)
   4290             try:  # Is 'c' acceptable as PathCollection facecolors?
-> 4291                 colors = mcolors.to_rgba_array(c)
   4292             except (TypeError, ValueError) as err:

~/Desktop/HomeWork/SJSU/cs156/anaconda/lib/python3.8/site-packages/matplotlib/
 ↪colors.py in to_rgba_array(c, alpha)
    340     else:
--> 341         return np.array([to_rgba(cc, alpha) for cc in c])
    342

~/Desktop/HomeWork/SJSU/cs156/anaconda/lib/python3.8/site-packages/matplotlib/
 ↪colors.py in <listcomp>(.0)
    340     else:
--> 341         return np.array([to_rgba(cc, alpha) for cc in c])
    342

~/Desktop/HomeWork/SJSU/cs156/anaconda/lib/python3.8/site-packages/matplotlib/
 ↪colors.py in to_rgba(c, alpha)
    188     if rgba is None:  # Suppress exception chaining of cache lookup␣
 ↪failure.
--> 189         rgba = _to_rgba_no_colorcycle(c, alpha)
    190         try:

~/Desktop/HomeWork/SJSU/cs156/anaconda/lib/python3.8/site-packages/matplotlib/
 ↪colors.py in _to_rgba_no_colorcycle(c, alpha)
    262     if not np.iterable(c):
--> 263         raise ValueError(f"Invalid RGBA argument: {orig_c!r}")
    264     if len(c) not in [3, 4]:

ValueError: Invalid RGBA argument: 0.5583269125217224

The above exception was the direct cause of the following exception:

ValueError                                Traceback (most recent call last)
<ipython-input-91-5a8050af201d> in <module>
     13 plt.title("Predicted labels")
     14 plt.scatter(X,y, s=25, c=y, marker=".")
---> 15 plt.scatter(X,y, s=20, c=predictions, marker="v")
```

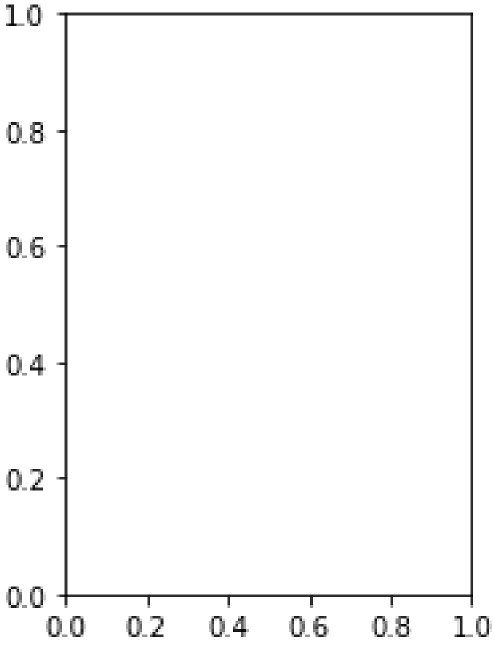
```
     16 plt.title("Predicted labels")
     17

~/Desktop/HomeWork/SJSU/cs156/anaconda/lib/python3.8/site-packages/matplotlib/
 ↪pyplot.py in scatter(x, y, s, c, marker, cmap, norm, vmin, vmax, alpha,␣
 ↪linewidths, verts, edgecolors, plotnonfinite, data, **kwargs)
   2888         verts=cbook.deprecation._deprecated_parameter,
   2889         edgecolors=None, *, plotnonfinite=False, data=None, **kwargs):
-> 2890     __ret = gca().scatter(

   2891         x, y, s=s, c=c, marker=marker, cmap=cmap, norm=norm,
   2892         vmin=vmin, vmax=vmax, alpha=alpha, linewidths=linewidths,

~/Desktop/HomeWork/SJSU/cs156/anaconda/lib/python3.8/site-packages/matplotlib/
 ↪__init__.py in inner(ax, data, *args, **kwargs)
   1436     def inner(ax, *args, data=None, **kwargs):
   1437         if data is None:
-> 1438             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1439
   1440         bound = new_sig.bind(ax, *args, **kwargs)

~/Desktop/HomeWork/SJSU/cs156/anaconda/lib/python3.8/site-packages/matplotlib/
 ↪cbook/deprecation.py in wrapper(*inner_args, **inner_kwargs)
    409                          else deprecation_addendum,
    410                 **kwargs)
--> 411         return func(*inner_args, **inner_kwargs)
    412
    413     return wrapper

~/Desktop/HomeWork/SJSU/cs156/anaconda/lib/python3.8/site-packages/matplotlib/
 ↪axes/_axes.py in scatter(self, x, y, s, c, marker, cmap, norm, vmin, vmax,␣
 ↪alpha, linewidths, verts, edgecolors, plotnonfinite, **kwargs)
   4449
   4450         c, colors, edgecolors = \
-> 4451             self._parse_scatter_color_args(

   4452                 c, edgecolors, kwargs, x.size,
   4453                 get_next_color_func=self._get_patches_for_fill.
 ↪get_next_color)

~/Desktop/HomeWork/SJSU/cs156/anaconda/lib/python3.8/site-packages/matplotlib/
 ↪axes/_axes.py in _parse_scatter_color_args(c, edgecolors, kwargs, xsize,␣
 ↪get_next_color_func)
   4295                 else:
   4296                     if not valid_shape:
-> 4297                         raise invalid_shape_exception(c.size, xsize)␣
 ↪from err
   4298                     # Both the mapping *and* the RGBA conversion failed␣
 ↪pretty
```

```
  4299                          # severe failure => one may appreciate a verbose␣
↪feedback.

ValueError: 'c' argument has 10 elements, which is inconsistent with 'x' and 'y ␣
↪with size 50.
```



[92]:
```
X1 = np.random.normal(loc=0, scale=3.0, size=int(n/2))
X2 = np.random.normal(loc=-3, scale=1, size=int(n/2))
X3 = np.random.normal(loc=-1, scale=1, size=int(n/2))
X, y, z = X1, X2, X3
X_train, X_test, y_train, y_test, z_train, z_test = train_test_split(X, y, z,␣
↪test_size=0.2, random_state=4)


plt.subplot(1, 2, 1)
plt.scatter(X,y,z, s=50, c=y, marker=".")
plt.scatter(X,y,z, s=25, c=y, marker="v")
plt.title("Actual labels")

plt.subplot(1, 2, 2)
plt.title("Predicted labels")
plt.scatter(X,y,z, s=25, c=y, marker=".")
plt.scatter(X,y,z, s=20, c=predictions, marker="v")
plt.title("Predicted labels")
plt.tight_layout()
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-92-c448b02827ce> in <module>
      7 
      8 plt.subplot(1, 2, 1)
----> 9 plt.scatter(X,y,z, s=50, c=y, marker=".")
     10 plt.scatter(X,y,z, s=25, c=y, marker="v")
     11 plt.title("Actual labels")

TypeError: scatter() got multiple values for argument 's'
```



[ ]:

[ ]:

[ ]: