a07

October 12, 2024

1 Assignment 7

[61]:
```
# Usage example:
# data = [((0, 0), 'a'), ((0, 1), 'a'), ((1, 0), 'a'), ((1, 1), 'a'), ((3, 4),␣
 ↪'b'), ((3, 5), 'b'), ((4, 4), 'b'), ((4, 5), 'b')]
# q = (1, 2)
# k = len(data) // 2
def knn_pred(k, data, q):
    dists = []
    for p in data:
        dist_sq = 0
        for i in range(len(q)):
            dist_sq += (q[i] - p[0][i]) ** 2
        dists.append((dist_sq ** 0.5, p[1]))
    dists.sort(key=lambda x: x[0])
    vote = {}
    for i in range(k):
        if not dists[i][1] in vote:
            vote[dists[i][1]] = 0
        vote[dists[i][1]] += 1
    pred = None
    for key in vote.keys():
        if pred == None:
            pred = key
            continue
        if vote[key] > vote[pred]:
            pred = key
    return pred

def best_k(k_vals, train, val):
    best_correct = 0
    for k in k_vals:
        correct = 0
        for p in val:
            if knn_pred(k, train, p[0]) == p[1]:
                correct += 1
        if correct > best_correct:
```

```
            best_k = k
            best_correct = correct
    return best_k
```

1.1 Question 1: Action/Comedy Classification

[62]:
```
train = [
    ((100, 0), 'Comedy'),
    ((0, 100), 'Action'),
    ((15, 90), 'Action'),
    ((85, 20), 'Comedy'),
]

val = [
    ((10, 95), 'Action'),
    ((85, 15), 'Comedy')
]

k_vals = range(1, len(train) + 1, 2)
k = best_k(k_vals, train, val)
k
```

[62]:
```
1
```

[63]:
```
test = [
    (6, 70),
    (93, 23),
    (50, 50),
]

test_pred = [knn_pred(k, train, x) for x in test]
test_pred
```

[63]:
```
['Action', 'Comedy', 'Comedy']
```

1.2 Question 2: Iris Dataset

[64]:
```
from sklearn import datasets

iris = datasets.load_iris()

test_split = 0.3
val_split = 0.3

train = []
val = []
```

```
test = []

num_class = len(iris['target_names'])
num_class_rec = len(iris['data']) // num_class

for i in range(num_class):
    for j in range(num_class_rec):
        feats = iris['data'][i * num_class_rec + j]
        label = iris['target'][i * num_class_rec + j]
        if j < test_split * num_class_rec:
            test.append((feats, label))
        elif j < test_split * num_class_rec + val_split * (1 - test_split) *␣
 ↪num_class_rec:
            val.append((feats, label))
        else:
            train.append((feats, label))

k_vals = range(1, 10)
k = best_k(k_vals, train, val)
k
```

[64]: 9

[65]:
```
test_pred = [knn_pred(k, train, x[0]) for x in test]

correct = 0
for i in range(len(test)):
    if test_pred[i] == test[i][1]:
        correct += 1

print(f'Accuracy: {correct / len(test)} ({correct}/{len(test)})')
```

```
Accuracy: 0.9555555555555556 (43/45)
```