

## 04CustomDatasets

**command**

11. *Microtus pennsylvanicus*

polymeric carbon density

```

import numpy as np
import sys
import random
import time

def read_data(filename):
    data = np.loadtxt(filename)
    return data

def split_data(data):
    train_data, test_data = data[0:10000], data[10000:]
    return train_data, test_data

def train_model(train_data, test_data):
    # Training
    model = LogisticRegression()
    model.fit(train_data)

    # Testing
    accuracy = model.score(test_data)
    return accuracy

if __name__ == '__main__':
    data = read_data('data.csv')
    train_data, test_data = split_data(data)
    accuracy = train_model(train_data, test_data)
    print('Accuracy: %.2f' % accuracy)

```

```

3) Processing data
import pandas as pd
from sklearn.preprocessing import StandardScaler

data = pd.read_csv('data.csv')
data = data.dropna()
data = data.reset_index(drop=True)

scaler = StandardScaler()
data[['feature1', 'feature2']] = scaler.fit_transform(data[['feature1', 'feature2']])

```

[illegible]

```

def merge_sort(a: list, b: list) -> list:
    """Merge sort algorithm"""
    if len(a) < 2:
        return a
    mid = len(a) // 2
    left = merge_sort(a[:mid])
    right = merge_sort(a[mid:])
    return merge(left, right)

def merge(left: list, right: list) -> list:
    """Merge two sorted lists"""
    result = []
    i = j = 0
    while i < len(left) and j < len(right):
        if left[i] <= right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
            j += 1
    result += left[i:]
    result += right[j:]
    return result

```