Thomas Dinh

DSE 6211

Week 3

Sept 14, 2023

1) After working through all of the code in the lab, run 'head(training_features)' and 'head(test_features)'. Copy and paste the output.

```
head(training_features)

```
```

          [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]   [,10]
[1,] -1.6331362 -0.5589410 -0.9639923 -0.4124804 -0.4926990 -1.0487310 0.69932583 0.29601744 -0.2903577 2.6848994
[2,] -0.2937300 -0.6128508 -0.2856842 -1.1284038 -0.7510443 1.3092785 0.92574384 -0.11474829 -0.7612385 2.9988251
[3,] -0.5078898 -0.7386406 0.7996087 -0.8458025 -0.4582530 0.3523676 0.96348018 -1.29569978 -1.5721999 0.2108298
[4,] -0.7873865 -0.2893915 0.6639471 -1.1284038 -0.7854904 0.2573230 1.34084353 -0.06340257 -1.2582794 0.1994005
[5,] -1.3354904 -1.3226645 -0.8283307 -0.8693526 -0.8888285 -1.2989503 -0.05540089 0.09063458 0.2328432 -0.5267431
[6,] -1.3173412 1.5525299 -0.9639923 -1.1284038 -0.8027134 -1.1392496 -0.31955524 0.50140031 0.6775640 -0.3720661
         [,11]   [,12]   [,13]   [,14]   [,15]   [,16]   [,17]   [,18]   [,19]   [,20]
[1,] -0.2249868 -0.8987075 -0.256638 -0.04463561 -0.03712769 -0.5005831 -0.08689792 -0.2320273 -0.2189171 -0.3128936
[2,] -0.2249868 -0.8987075 -0.256638 -0.04463561 -0.03712769 -0.5005831 -0.08689792 4.3091780 -0.2189171 -0.3128936
[3,] -0.2249868 -0.8987075 -0.256638 -0.04463561 -0.03712769 1.9973648 -0.08689792 -0.2320273 -0.2189171 -0.3128936
[4,] -0.2249868 -0.8987075 -0.256638 -0.04463561 -0.03712769 1.9973648 -0.08689792 -0.2320273 -0.2189171 -0.3128936
[5,] -0.2249868 -0.8987075 -0.256638 -0.04463561 -0.03712769 1.9973648 -0.08689792 -0.2320273 -0.2189171 -0.3128936
[6,] -0.2249868 -0.8987075 -0.256638 -0.04463561 -0.03712769 -0.5005831 -0.08689792 -0.2320273 -0.2189171 -0.3128936
         [,21]   [,22]   [,23]   [,24]   [,25]   [,26]   [,27]   [,28]   [,29]   [,30]
[1,] -0.2924153 -0.05105843 -0.05810599 -0.01749279 -0.01749279 -0.1599022 -0.1473846 -0.1429911 -0.1289892 -0.05105843
[2,] -0.2924153 -0.05105843 -0.05810599 -0.01749279 -0.01749279 -0.1599022 -0.1473846 -0.1429911 -0.1289892 -0.05105843
[3,] -0.2924153 -0.05105843 -0.05810599 -0.01749279 -0.01749279 -0.1599022 -0.1473846 -0.1429911 -0.1289892 -0.05105843
[4,] -0.2924153 -0.05105843 -0.05810599 -0.01749279 -0.01749279 -0.1599022 -0.1473846 -0.1429911 -0.1289892 -0.05105843
[5,] -0.2924153 -0.05105843 -0.05810599 -0.01749279 -0.01749279 -0.1599022 -0.1473846 -0.1429911 -0.1289892 -0.05105843
[6,] -0.2924153 -0.05105843 -0.05810599 -0.01749279 -0.01749279 -0.1599022 -0.1473846 -0.1429911 -0.1289892 -0.05105843
         [,31]   [,32]   [,33]
[1,] -0.1105939 -0.01749279 -0.04795369
[2,] -0.1105939 -0.01749279 -0.04795369
[3,] -0.1105939 -0.01749279 -0.04795369
[4,] -0.1105939 -0.01749279 -0.04795369
[5,] -0.1105939 -0.01749279 -0.04795369
[6,] -0.1105939 -0.01749279 -0.04795369
```

```
head(test_features)

```
```
            [,1]        [,2]       [,3]         [,4]        [,5]       [,6]       [,7]        [,8]       [,9]
[1,] -0.82731459 -0.19055667  0.2569622 -0.195819349  0.55790517 -0.4875157  1.1898982  0.39870888 -0.8135586
[2,] -0.37358621 -0.26243653 -0.8283307 -0.007418448 -0.68215225  0.6103456  0.7747985  0.60409174 -0.2118776
[3,] -0.02512282  0.05203785 -0.1500226  0.567204303  0.07566062  0.4318946  0.7747985  0.96351176 -0.1333974
[4,] -0.16305625  0.66301665 -0.2856842 -0.954132979 -0.68215225  1.7010947 -0.3950279  1.42562321  1.1222848
[5,] -0.17394573  1.49862002  0.2569622 -0.822252348 -0.56159111  1.5776015 -1.3007000 -0.01205686  1.0961247
[6,] -0.40262483 -0.83747541  0.7996087  1.447978519  2.22853807 -1.1256718  0.7370622 -1.39839121 -1.4152397
          [,10]      [,11]      [,12]     [,13]       [,14]       [,15]      [,16]       [,17]      [,18]      [,19]
[1,] -0.8322873 -0.2249868 -0.8987075 -0.256638 -0.04463561 -0.03712769 -0.5005831 -0.08689792  4.3091780 -0.2189171
[2,]  0.8104888 -0.2249868 -0.8987075 -0.256638 -0.04463561 -0.03712769 -0.5005831 -0.08689792 -0.2320273 -0.2189171
[3,] -0.2760597 -0.2249868 -0.8987075 -0.256638 -0.04463561 -0.03712769 -0.5005831 -0.08689792 -0.2320273 -0.2189171
[4,] -0.3530172 -0.2249868 -0.8987075 -0.256638 -0.04463561 -0.03712769 -0.5005831 -0.08689792  4.3091780 -0.2189171
[5,] -0.1518610 -0.2249868 -0.8987075 -0.256638 -0.04463561 -0.03712769 -0.5005831 -0.08689792 -0.2320273 -0.2189171
[6,]  0.3594112 -0.2249868 -0.8987075 -0.256638 -0.04463561 -0.03712769 -0.5005831 -0.08689792 -0.2320273 -0.2189171
          [,20]      [,21]       [,22]       [,23]       [,24]       [,25]      [,26]      [,27]      [,28]      [,29]
[1,] -0.3128936 -0.2924153 -0.05105843 -0.05810599 -0.01749279 -0.01749279 -0.1599022 -0.1473846 -0.1429911 -0.1289892
[2,] -0.3128936 -0.2924153 -0.05105843 -0.05810599 -0.01749279 -0.01749279 -0.1599022 -0.1473846 -0.1429911 -0.1289892
[3,] -0.3128936 -0.2924153 -0.05105843 -0.05810599 -0.01749279 -0.01749279 -0.1599022 -0.1473846 -0.1429911 -0.1289892
[4,] -0.3128936 -0.2924153 -0.05105843 -0.05810599 -0.01749279 -0.01749279 -0.1599022 -0.1473846 -0.1429911 -0.1289892
[5,] -0.3128936 -0.2924153 -0.05105843 -0.05810599 -0.01749279 -0.01749279 -0.1599022 -0.1473846 -0.1429911 -0.1289892
[6,] -0.3128936 -0.2924153 -0.05105843 -0.05810599 -0.01749279 -0.01749279 -0.1599022 -0.1473846 -0.1429911 -0.1289892
           [,30]      [,31]       [,32]       [,33]
[1,] -0.05105843 -0.1105939 -0.01749279 -0.04795369
[2,] -0.05105843 -0.1105939 -0.01749279 -0.04795369
[3,] -0.05105843 -0.1105939 -0.01749279 -0.04795369
[4,] -0.05105843 -0.1105939 -0.01749279 -0.04795369
[5,] -0.05105843 -0.1105939 -0.01749279 -0.04795369
[6,] -0.05105843 -0.1105939 -0.01749279 -0.04795369
```

2) What is the rank of the tensor 'training_features'? What is the shape of 'training_features'? How many dimensions does 'training_features' have along the second axis?

The rank of a training feature is a 2D tensor or a rank 2. The shape of the training_feature is (6537, 33). It has dimension along the second axis of 33.

3) State two situations where scaling the numerical variables is important.

Feature scaling is crucial for machine learning algorithms that rely on calculating distances between data points. These algorithms are sensitive to the relative scales of features. For example, consider two features: 'weight' and 'distance.' These features have different units and magnitudes; weight could be measured in pounds (e.g., 15 pounds), while distance could be measured in feet (e.g., 5300 ft). Without proper scaling, the algorithm might treat these features as having equal importance, potentially leading to biased results. Feature scaling rectifies this issue by bringing all features to a common scale. Additionally, scaling contributes to faster convergence in neural networks and enhances the training process. When algorithms compute distances or assume normality, feature scaling becomes essential.

Another reason for feature scaling is that certain algorithms are particularly sensitive to feature magnitudes. Some of these algorithms include:

- K-nearest neighbors (KNN): KNN, which uses a Euclidean distance measure, is sensitive to magnitudes and should have all features scaled to ensure equal weighting.
- K-Means: K-Means clustering also employs the Euclidean distance measure, making feature scaling crucial in this context.
- Principal Component Analysis (PCA): PCA aims to capture features with the maximum variance. Therefore, scaling is necessary to prevent skewing towards high-magnitude features.