Report

# Clustering

Report Task 1

The K-means algorithm separates samples in N number of groups of equal variances by minimizing the sum of squared distances between each data point, also known as the inertia. Given that our Covertype dataset contains high-dimensional data, this puts some limitations on this model. The Euclidean distances between data points tend to become uniformly large [1]. To tackle this problem, I reduced the dimensionality of the data using PCA. I used the “mle” parameter which determines the most optimal number of principal components. Additionally, outliers can significantly affect K-means because the cluster centroids are determined by the mean of all the points assigned to the cluster. This means outliers can skew the centroids and potentially increase the error for points in actual dense regions. That is why I apply scale to the dimensional reduced data with RobustScaller which is robust to outliers. Then I apply KMeans on the scaled data with initialization – “init” parameter set to “k-means++” which initializes the centroids to be distant from each other and set the number of initializations counts to 10 for better initial centroids.

I applied GaussianMixture with a covariance type “tied” because the full covariance matrix can lead to overfitting given the high dimensionality of the data.

Report Task 2

The K-means model made *431363* (2.24%) errors, GMM made *210538* (1.09%) and the random assigning model made *16466471* (85.71%) errors out of 19210967 total pairs of data points with the same label.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **K-means** | **Gaussian mixture model** | **Random baseline** |
| **Error count** | 431363 | 210538 | 16466471 |
| **Total pairs with the same label** | 19210967 | 19210967 | 19210967 |
| **Accuracy percentage** | 97.76% | 98.91% | 14.29% |

Report Task 3

The high accuracy of the K-means model indicates that the cluster shapes are approximately spherical, and that is why we can’t notice a significant improvement in the Gaussian Mixture model, although this model has the ability to capture variations in the data that K-means isn’t able to. The random baseline has a very low accuracy which highlights that the k-means and GMM models are capturing meaningful patterns and are far better than random guessing.

# Classification

Report Task 4

Support vector machines require extensive training time for large datasets like the Covertype dataset. For non-linear SVMs usually the Gram matrix is computed making the training time scale at least quadratically with the number of samples which becomes impractical beyond the thousands of data points [2]. The LinearSVC is a faster implementation of a Support Vector Machine in the case of a linear kernel. It uses liblinear algorithm which avoids computing the Gram matrix and scales linearly to the number of samples. The disadvantage of this model is the inability to capture non-linear patterns within the data with an accuracy of around 70%.

Report Task 5

Figure 1 - The graph illustrates the relationship between the maximum depth of the decision tree and the accuracy of the model on the validation dataset.

As a decision tree requires little data preparation no scaling or dimension reduction is applied to the data. Initially, I ran the DecisionTreeClassifier with its default parameters without any constraints on the size of the tree with gini as the function for measuring the quality of the split. The accuracy score on the same training set was 100% which clearly stated that the model is overfitting the training data. Applying PCA gave the same result. After that, I created models with decreasing maximum depth from 5 to 45 on every 5 layers. Figure 1 shows the results from the grid search which are obtained with cross-validation. The graph shows that the accuracy doesn’t increase substantially for trees bigger than 25 layers. The same approach was used for the criterion (the function that measures the quality of the split when training the model) but all models showed similar results.

|  |  |  |
| --- | --- | --- |
| lbfgs | 61.23 | NA |
| liblinear | 70.48 | 68.01 |
| newton-cg | 69.06 | NA |
| newton-cholesky | 71.55 | NA |
| sag | 67.86 | NA |
| saga | 68.40 | 67.60 |
|  | L2 | L1 |

Report Task 6

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Logistic Regression** | **Decision Trees** | **Ensemble method** |
| **Accuracy percentage** |  | 93.04% |  |

# Regression

Report Task 7

When we plot the data, we can notice that the data has some polynomial features. The 2 curves indicate that the function should be of degree 3, but I generated polynomial features from degrees 1 to 5 and validated them using cross-validation.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Linear Regression** | **Neural Network** | **Bayesian** |
| **Accuracy percentage** |  |  |  |