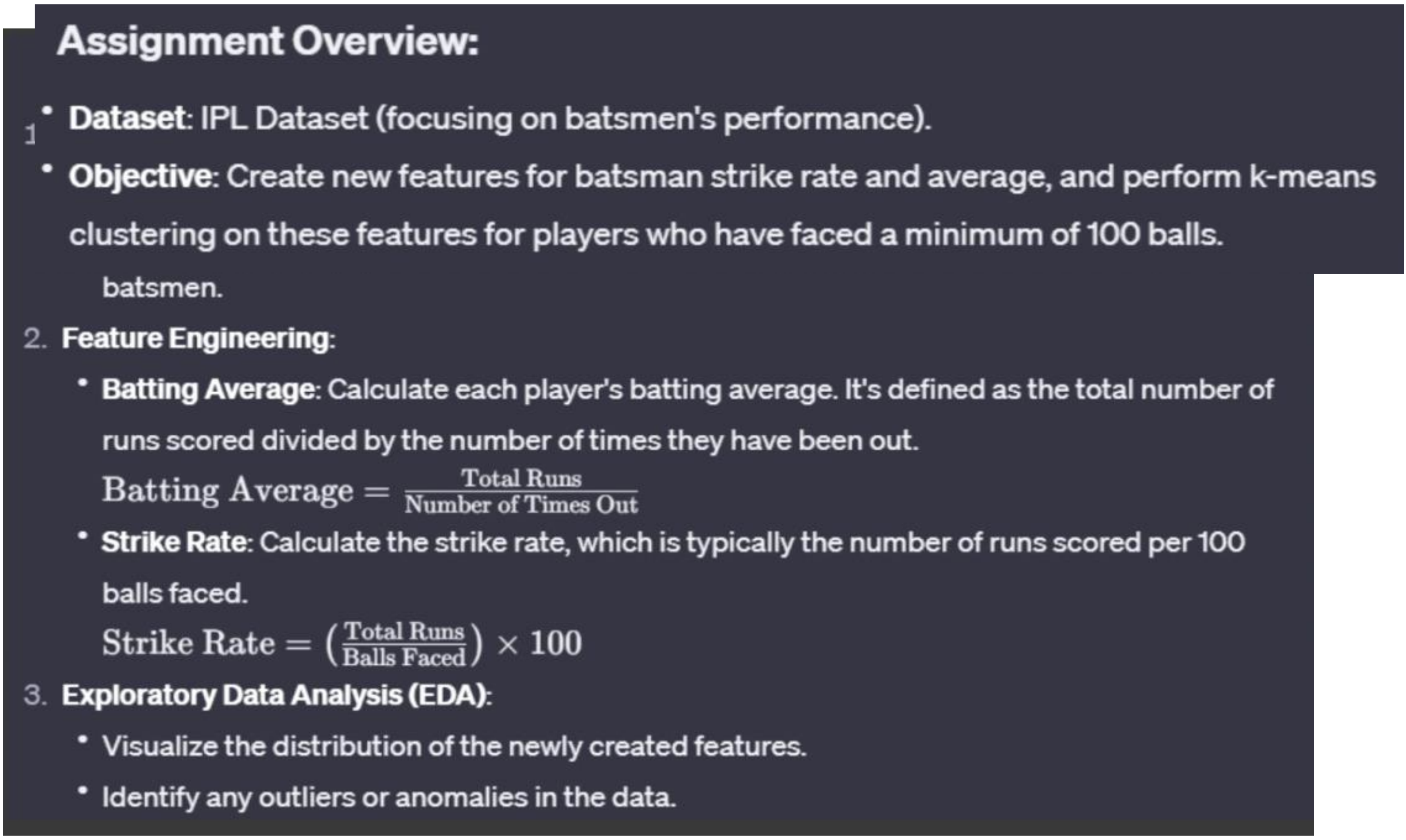
Recap

28 November 2023 19:14

Assignment Solution

28 November 2023 19:14



Assignment Overview:

1

- **Dataset**: IPL Dataset (focusing on batsmen's performance).
- **Objective**: Create new features for batsman strike rate and average, and perform k-means clustering on these features for players who have faced a minimum of 100 balls.

batsmen.

2. **Feature Engineering**:
   - **Batting Average**: Calculate each player's batting average. It's defined as the total number of runs scored divided by the number of times they have been out.
     $\text{Batting Average} = \frac{\text{Total Runs}}{\text{Number of Times Out}}$
   - **Strike Rate**: Calculate the strike rate, which is typically the number of runs scored per 100 balls faced.
     $\text{Strike Rate} = \left(\frac{\text{Total Runs}}{\text{Balls Faced}}\right) \times 100$
3. **Exploratory Data Analysis (EDA)**:
   - Visualize the distribution of the newly created features.
   - Identify any outliers or anomalies in the data.

Silhouette Score

28 November 2023 19:15

Cohesion

Definition: Cohesion refers to the degree to which elements within the same cluster are close to each other. It measures how tightly grouped the data points in a cluster are.

Ideal Scenario: High cohesion means that data points in a cluster are similar or near to each other, indicating a good clustering where each cluster is distinct and meaningful.

Measurement: Cohesion can be quantified using metrics such as the sum of squared distances of data points from their respective cluster centroids. In K-Means, this is often referred to as inertia or within-cluster sum of squares.

Separation

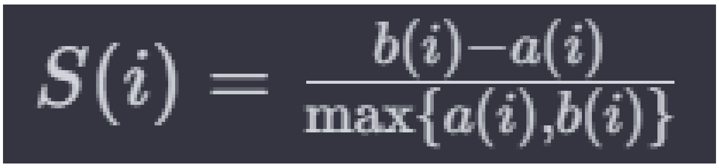
Definition: Separation, on the other hand, refers to how distinct or well-parted different clusters are from each other. It measures the extent to which clusters are different or distant from each other.

Ideal Scenario: High separation means that clusters are well-differentiated and far apart, indicating that the algorithm has done a good job in distinguishing between different groups in the data.

Measurement: Separation can be quantified by metrics such as the distance between cluster centroids, or more complex measures like the silhouette score, which considers both cohesion and separation.

Silhouette Score

The silhouette score is a measure used to assess the quality of clusters created by a clustering algorithm. It provides a succinct graphical representation of how well each data point lies within its cluster, which is a combination of both cohesion and separation. The value of the silhouette score ranges from -1 to + 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters.



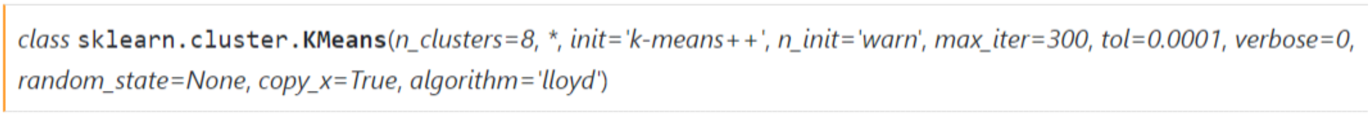
$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Interpretation

Close to +1: Indicates that the data point is far away from the neighboring clusters.
Close to 0: Indicates that the data point is on or very close to the decision boundary between two neighboring clusters.
Close to -1: Indicates that the data point may have been assigned to the wrong cluster.

Kmeans Hyperparameters

28 November 2023 19:16



```
class sklearn.cluster.KMeans(n_clusters=8, *, init='k-means++', n_init='warn', max_iter=300, tol=0.0001, verbose=0,
random_state=None, copy_x=True, algorithm='lloyd')
```

Number of Clusters (k):

Description: This is the number of clusters you want the algorithm to form, as well as the number of centroids to generate.
Importance: Choosing the right number of clusters is crucial as it significantly influences the clustering results. Too many clusters can overfit the data, while too few can miss important patterns.

Initialization Method:

Description: This parameter specifies the method for initializing the centroids. Common methods include 'random' (randomly choosing k data points as initial centroids) and 'k-means++' (a smarter way of initializing centroids to improve convergence).
Importance: Good initialization can lead to faster convergence and better clustering. K-means++ is generally preferred over random initialization.

Number of Initialization Runs (n_init):

Description: This is the number of times the KMeans algorithm will be run with different centroid initializations. The final results will be the best output of n_init consecutive runs in terms of inertia.
Importance: Multiple initializations can prevent the algorithm from falling into sub-optimal solutions, but increase computational cost.

Maximum Number of Iterations (max_iter):

Description: The maximum number of iterations the algorithm will run for each initialization.
Importance: A higher number of iterations allows more time for convergence but increases computational time. Usually, KMeans converges well before reaching the maximum number of iterations.

Tolerance (tol):

Description: This is the tolerance to declare convergence. If the centroids do not move significantly (as defined by this parameter) in consecutive iterations, the algorithm stops.
Importance: A smaller tolerance can lead to a more precise solution, but might increase computation time. A larger tolerance might speed up the algorithm but can lead to less precise clustering.

Algorithm:

Description: The computational algorithm to use. Options typically include 'auto', 'full', or 'Elkan'. 'Full' corresponds to the classic EM-style algorithm, while 'elkan' is an optimized variant that is generally more efficient.

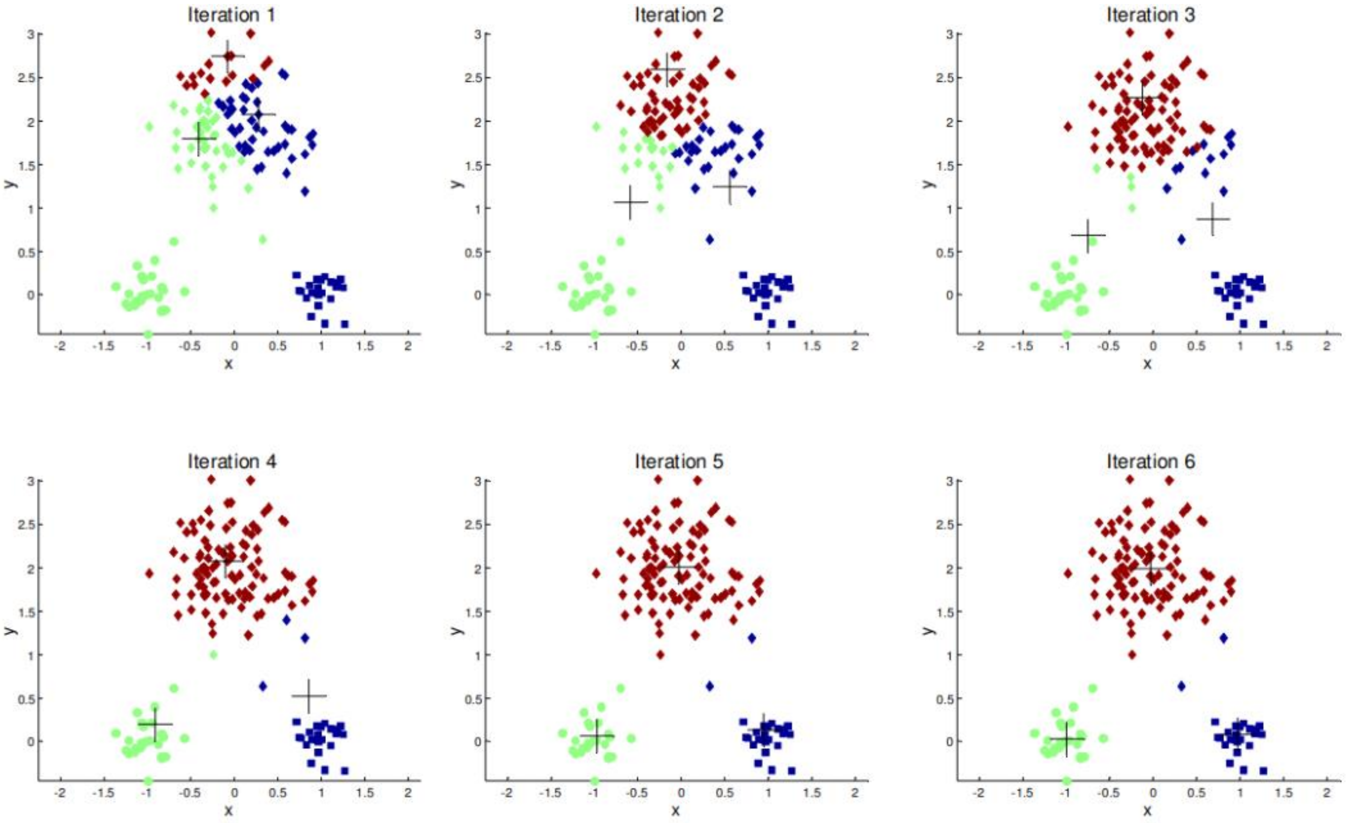
Importance: The choice of algorithm affects the computational efficiency. 'Elkan' is often faster but works only with Euclidean distance.

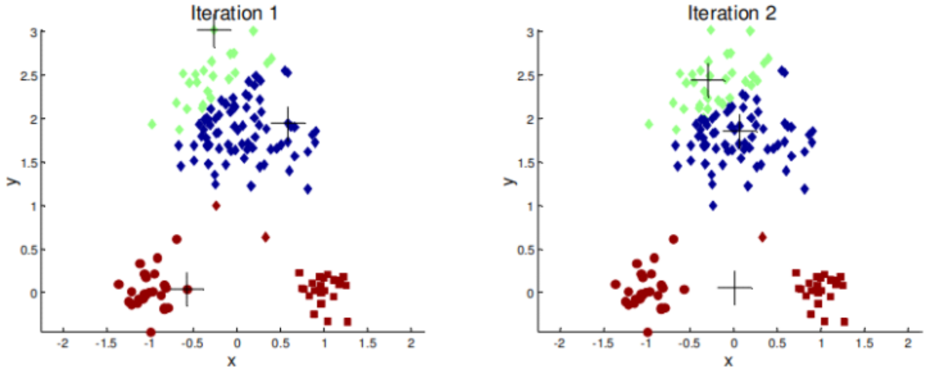
Random State:

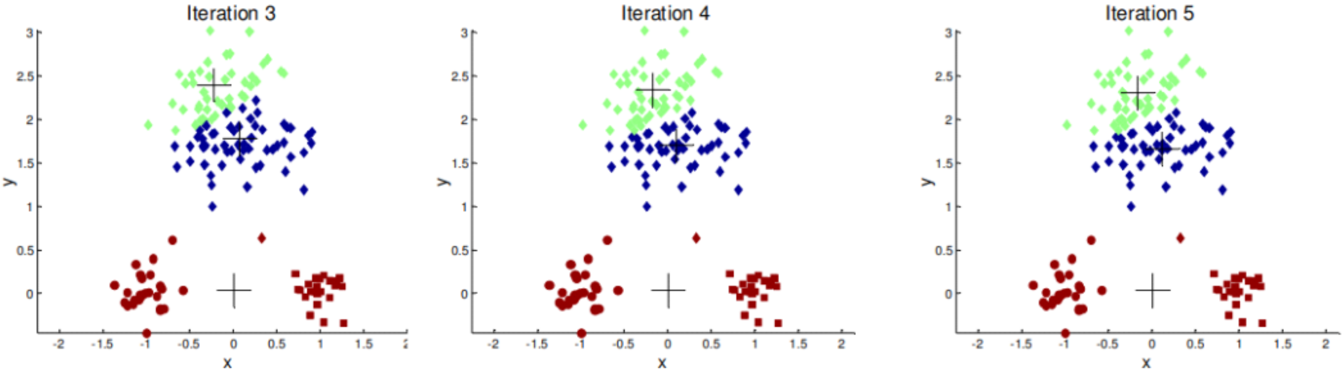
Description: Determines the random number generation for centroid initialization, which influences the behavior of the algorithm.
Importance: Setting a random state ensures reproducibility, as KMeans can produce different results on different runs due to its stochastic nature.

Kmeans++

28 November 2023 19:16







KMeans++ is an algorithm for choosing the initial values (or "seeds") for the KMeans clustering algorithm. The standard KMeans algorithm is sensitive to the initial starting points (centroids), and KMeans++ provides a way to overcome this problem by specifying a procedure to initialize the centroids before proceeding with the standard KMeans iterative algorithm.

Here's a simplified overview of how KMeans++ works:

Initial Centroid Selection: The first centroid is chosen uniformly at random from the data points that are being clustered.

Distance Calculation: Calculate the distance of each data point from the nearest, previously chosen centroid.

Probabilistic Selection of Next Centroids: Choose the next centroid from the data points with a probability proportional to the square of the distance from the point to its nearest centroid. This step biases the algorithm to select data points that are far from the existing centroids.

Repeat Until K Centroids: Repeat steps 2 and 3 until k centroids have been chosen.

Proceed with Standard KMeans: Once the initial centroids are chosen, proceed with the standard KMeans clustering algorithm.

This method tends to spread out the initial centroids, which can lead to better clustering results compared to selecting the initial centroids randomly, as the standard KMeans algorithm does. By doing so, KMeans++ can often lead to faster convergence and better clustering.

Accelerated Kmeans

28 November 2023 19:16

Elkan KMeans is an optimized version of the standard KMeans algorithm, known for its computational efficiency, particularly in certain types of datasets. Here are five summary points about Elkan KMeans:

Triangle Inequality Optimization: Elkan KMeans utilizes the triangle inequality theorem to reduce the number of distance calculations required. This theorem states that in a triangle, the length of any side is less than the sum of the other two sides. This property is used to avoid unnecessary distance calculations, thereby speeding up the algorithm.

Efficient for Dense Data: This optimization is particularly effective for datasets with dense clusters. By minimizing distance calculations in such environments, Elkan KMeans can significantly outperform the standard KMeans algorithm in terms of computational time.

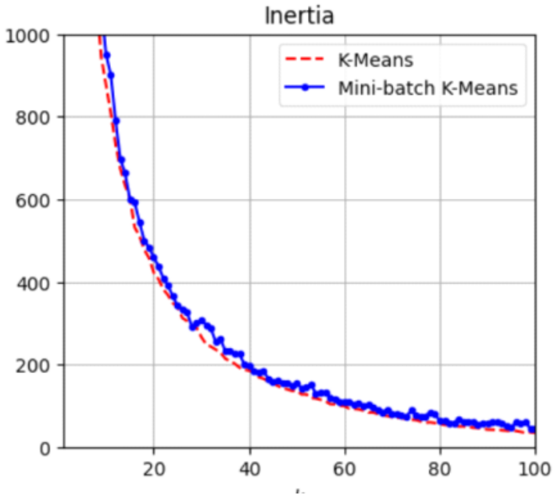
Lower Bound Calculations: Elkan KMeans maintains and updates lower bounds for distances between data points and centroids. These bounds help in quickly determining if a point could potentially belong to a different cluster without explicitly calculating the distance, further improving efficiency.

Improved Convergence Rate: Due to the reduced number of distance calculations and efficient updating of clusters, Elkan KMeans often converges faster than the standard algorithm, especially as the number of dimensions and the size of the dataset increase.

Dependence on Euclidean Distance: A limitation of Elkan KMeans is that it's specifically optimized for Euclidean distances. This means that its optimizations are not applicable if a different distance metric is used, which can be a constraint in certain applications.

Mini Batch Kmeans

28 November 2023 19:16



Assignment

28 November 2023 19:17

Reading Assignment

28 November 2023 19:17