Majadbi_Yousef_CSE511_Hot Spot Analysis Project Report.

Reflection

For this project, I performed spatial hot spot analysis on a large dataset of NYC taxi trips using Apache Spark and Scala. To give the client access to statistically significant geographic locations which it can utilize to plan its operations in advance and import customer service.

I started with setting up the environment, including the Apache Spark, Scala and other required tools.

I completed two different hot spot analysis tasks: Hot Zone Analysis and Hot Cell Analysis. For the Hot Zone Analysis, I modified the *HotzoneUtils.scala* and *HotzoneAnalyiss.scala* files. And for Hot Cell task, I modified the *HotcellUtils.scala* and *HotcellAnalysis.scala* files.

Lessons Learned

Through this project I learned the following:

1. How to use **Apache Spark** for big data processing and particularly for spatial data processing.
2. How to perform range join operations, Spatial queries and Statistical analysis using **Scala**.
3. How to set up and configure a big data processing environment on Windows.

Implementation

Hot Zone Analysis:

In this task, I performed a range join operation on a rectangle dataset and a point dataset. to determine the hotness of each rectangle based on the number of points it contains. I implemented the *ST_Contains* function to check if a point lies within a given rectangle. The function takes two strings as input: the corner points of the rectangle and the point. It returns a Boolean indicating whether the point is within the rectangle.

```
def ST_Contains(queryRectangle: String, pointString: String): Boolean = {
  val rectCoords = queryRectangle.split(",")
  val rectX1 = rectCoords(0).trim.toDouble
  val rectY1 = rectCoords(1).trim.toDouble
  val rectX2 = rectCoords(2).trim.toDouble
  val rectY2 = rectCoords(3).trim.toDouble
```

```
    val pointCoords = pointString.split(",")
    val pointX = pointCoords(0).trim.toDouble
    val pointY = pointCoords(1).trim.toDouble

    val minX = math.min(rectX1, rectX2)
    val maxX = math.max(rectX1, rectX2)
    val minY = math.min(rectY1, rectY2)
    val maxY = math.max(rectY1, rectY2)

    if (pointX >= minX && pointX <= maxX && pointY >= minY && pointY <= maxY) {
      return true
    }
    false
  }
}
```

I then modified the provided code template to perform the range join operation and count the points within each rectangle. The results were sorted by the rectangle string in ascending order.

```
    val resultDf= joinDf.groupBy("rectangle").count().sort("rectangle").coalesce(1)

    resultDf.show()
    return resultDf
  }
```

Hot Cell Analysis:

In this task I focused on applying spatial statistics to spatio-temporal big data in order to identify statistically significant spatial hot spots using Apache Spark.

The steps required before calculating the z-score included:

1. Loading the data and determine the cell coordinates (x,y,z).
2. Calculating the number of points in each cell.
3. Computing the mean and standard deviation of the point counts.
4. Calculating the Getis_Ord stats for each cell.

The results were sorted by the G-score and the top 50 hottest cells were identified.

Conclusion

The Hot Spot Analysis project successfully identified significant geographic locations with high taxi activity using Apache Spark and Scala. The insights gained from this analysis can help improve operational planning and customer service for the taxi company. This project enhanced my understanding of big data processing and spatial analysis, providing valuable skills for future data science projects.