**Final Project Report**

Pavan Kalyan Imadabathini

Anushya Sivanupandian

Aasritha Devi Surapaneni

**Geospatial Analysis Using Apache Spark**

Group Name: Team Devils

IFT 512: Advanced Analytics Big Data/AI

Prof. Robert Rucker

Dec 3rd, 2023

**Introduction**

**A. Background and Motivation**

The area of focus for this project is the optimization of urban transportation systems through geospatial analysis of New York City's Yellow Cab taxi trip data. The motivation for selecting this area stems from the increasing challenges faced in urban mobility, particularly in densely populated cities like New York. Traffic congestion, inefficient taxi routing, and the underutilization of resources are prominent issues that significantly impact the daily lives of city dwellers and the environment.

In the context of urban planning and transportation management, the ability to analyze and interpret large-scale taxi trip data presents an opportunity to gain valuable insights into travel patterns, peak traffic times, and popular destinations. This understanding is crucial for devising strategies to enhance the efficiency of taxi operations, reduce traffic congestion, and improve overall urban mobility.

The project is driven by the need to leverage big data technologies for urban betterment. By employing tools like Apache Spark and Hadoop, combined with sophisticated data analysis techniques, the project aims to uncover patterns and trends that can inform smarter urban planning decisions. The goal is to provide actionable intelligence that can help taxi companies optimize their services, assist city planners in improving traffic management, and ultimately contribute to the creation of more livable, sustainable urban environments.

This approach aligns with the broader trend of smart city initiatives, where data-driven insights are used to improve various aspects of city living. In this case, the focus is on transforming the taxi industry and urban transportation landscape, which are integral components of the city's infrastructure.

Through this project, we aim to demonstrate the power of data analytics in solving real-world problems and contributing to the advancement of smart, sustainable urban living. The insights gained from the analysis of NYC's Yellow Cab data are expected to be a valuable resource for taxi companies, city planners, and policymakers in their efforts to enhance urban transportation systems.

1. **Hot-zone Analysis Objective**: To assess and quantify the 'hotness' of defined rectangular areas. This 'hotness' is determined by the quantity of points within each rectangle, with a higher number of points indicating a 'hotter' zone. Methodology: a. Ensuring that the input for queryRectangle and pointString is correctly formatted and non-empty. b. Extracting the geographical coordinates from these input strings. c. Determining whether each point falls within the boundaries of a given rectangle. d. Incrementally tallying the number of points in each rectangle to determine its relative 'hotness.'
2. **Hot-cell Analysis Objective**: To compute the Getis Ord statistic, a crucial metric in spatial data analysis, particularly applied to the NYC Taxi Trip dataset. Methodology: a. Collating monthly data of taxi trips. b. Pinpointing the top 50 hotspots on the Getis Ord (Gi\*) value. c. This involves calculating cell values based on coordinates, determining spatial weights, & computing the Gi\* value, a z-score that measures the hotspot significance.

A white background with black text

Description automatically generated

**Project Structure**

1. **Focus Questions to Address**

* How can analyzing taxi trip data inform better urban planning and traffic management strategies?

Analyzing taxi trip data is invaluable for urban planning and traffic management, as it provides insights into traffic flow, congestion patterns, and popular routes. This information aids in optimizing traffic signals, planning road expansions, and developing efficient public transportation routes. Additionally, it informs zoning decisions, commercial development, and environmental impact assessments. Such data also plays a crucial role in emergency and event planning, ensuring smooth traffic flow during critical times. Overall, taxi trip data enables urban planners and traffic managers to make data-driven decisions, enhancing urban mobility and sustainability.

* What insights can hotspot analysis provide to optimize taxi service allocation and routes?

Hotspot analysis of geospatial data significantly enhances taxi services by identifying high-demand areas for better vehicle allocation, optimizing routes to reduce travel time, and strategically scheduling services during peak periods. This approach improves both operational efficiency and customer satisfaction.

* How can cluster analysis of taxi trip data help in identifying and mitigating traffic congestion hotspots?

Cluster analysis of taxi trip data aids in identifying traffic congestion hotspots by pinpointing high-density areas and peak congestion times. This facilitates route optimization and informs traffic management policies, effectively mitigating traffic congestion.

* What business intelligence can be extracted from the data to give taxi companies a competitive edge?

Taxi companies can gain a competitive edge by using data to forecast demand, tailor dynamic pricing, understand customer behavior, and optimize operational efficiency. This intelligence enables strategic market expansion and targeted marketing, enhancing service efficiency and customer satisfaction.

* How can the findings enhance the efficiency and service quality of taxi operations in urban areas?

The findings can enhance the efficiency and service quality of taxi operations in urban areas by enabling more precise demand forecasting and resource allocation. By identifying high-demand areas and peak times, taxi companies can strategically position their fleet, reducing wait times for customers and improving response rates. Additionally, insights into popular routes and traffic patterns allow for more efficient route planning, leading to faster service and reduced fuel consumption. Overall, these data-driven strategies lead to a more customer-centric and efficient taxi service in urban environments.

1. **DAG**

A screenshot of a computer

Description automatically generated

**C.CRISP**

The CRISP-DM (Cross-Industry Standard Process for Data Mining) model is a structured approach to planning and executing data mining projects. It consists of six phases:

1. Business Understanding

2. Data Understanding

3. Data Preparation

4. Modeling

5. Evaluation

6. Deployment

For the project on optimizing urban transportation systems using geospatial analysis, the CRISP-DM workflow could be tailored as follows:

1. **Business Understanding**: Define the objectives and requirements from a business perspective. For this project, the goal is to optimize taxi operations by analyzing trip data to improve urban mobility. This phase would involve clarifying the project's objectives, such as reducing congestion, improving service allocation, and enhancing customer satisfaction.
2. **Data Understanding**: Collect initial data and proceed with activities to get familiar with the data, identify data quality issues, and discover first insights into the data. For the taxi trip analysis, this would involve gathering New York City Yellow Cab trip data, understanding the structure of the data, and identifying any anomalies or patterns.
3. **Data Preparation**: This phase involves all activities to construct the final dataset from the initial raw data. This could include cleaning the data, handling missing values, integrating various data sources, and transforming variables as needed for the analysis. For the project, this might also include spatial data processing to ensure that the data is suitable for geospatial analysis.
4. **Modeling**: In this phase, various modeling techniques are selected and applied. For geospatial analysis, this could involve spatial clustering, hotspot analysis, and predictive modeling. The models would be calibrated to the objectives defined in the business understanding phase.
5. **Evaluation**: Once models are developed, they need to be evaluated with respect to the business objectives. The evaluation phase involves assessing whether the models meet the business's initial objectives and needs, and identifying any new questions or requirements that the modeling process has uncovered.
6. **Deployment**: The deployment phase involves implementing the models into the business's decision-making process. For a geospatial analysis project, this could mean integrating the models into a dashboard or reporting tool for taxi companies and city planners to use. It might also involve automating certain processes, such as dynamic taxi dispatch based on real-time data analysis.

Throughout this workflow, it's crucial to maintain a loop back to previous steps as new insights, challenges, or data sources emerge. This iterative process ensures that the project remains aligned with business objectives and leverages the data as effectively as possible.

A diagram of a diagram

Description automatically generated

**Technology Description**

The New York City Yellow Cab Taxi Trip Records dataset encompasses data from January 2009 to June 2015 and offers an extensive array of records detailing taxi trips within the city. It incorporates specifics such as locations of pick-up and drop-off, time records, charges, and other pertinent data points. This dataset is pivotal for our investigation as it permits the execution of spatial queries and the identification of high-activity areas.

This dataset is derived from verified records, which guarantees the precision and applicability of the data for our spatial analysis endeavor. Data preprocessing will be undertaken to refine, structure, and modify the data appropriately for subsequent analysis.

In terms of technology, the project is set to employ:

1. Apache Spark will be the primary data processing framework, chosen for its ability to manage extensive geospatial datasets through parallel processing.
2. The Hadoop Distributed File System (HDFS) will function as the storage solution, offering resilience and scalability which are crucial for handling large volumes of data.
3. The Scala programming language will be used to develop custom functions and queries in SparkSQL, with its functional programming capabilities enhancing data processing tasks.
4. Amazon's EC2 service will supply the computational power required to deploy our Spark cluster, with the flexibility to adapt the infrastructure based on the project's computational demands.

Collectively, these technologies provide a potent and expandable architecture for the analysis of geospatial data, ensuring that the project can effectively process and analyze sizeable and intricate data sets to extract meaningful insights.

**Knowledge and Value Claims**

1. **Knowledge Claim**

**Efficiency in Resource Allocation**: The analysis likely revealed specific patterns in demand that were previously unknown, allowing for more precise placement of taxi services.

**Unexpected Traffic Patterns**: The project may have discovered that some areas expected to be in high demand were not, possibly due to shifts in urban dynamics, new transportation options, or infrastructure changes.

**Influence of External Factors:** It might have been found that factors like weather, local events, or urban development projects had a more significant impact on taxi demand and traffic than initially thought, leading to new strategies for adaptive service planning.

1. **Value Claims**

The findings from this project have multiple applications that can benefit various stakeholders:

1. **City Planners and Traffic Management Authorities**: The insights can inform better urban planning decisions, such as the placement of new taxi stands, the design of traffic flow patterns, and the timing of traffic signals, contributing to reduced congestion and more efficient transportation networks.
2. **Taxi Companies and Drivers:** The knowledge gained can enable taxi companies to optimize dispatching, improve route planning, and enhance service allocation, leading to increased profitability and improved customer satisfaction. Drivers can benefit from understanding peak times and locations, allowing them to position themselves strategically to maximize earnings.
3. **Public Transport Authorities**: Understanding taxi travel patterns can help in the planning and optimization of public transport services, such as bus and subway schedules, to complement taxi services during peak hours and events.
4. **Ride-Sharing Services and Urban Mobility Apps**: These services can integrate the insights to improve their algorithms for route optimization, carpooling suggestions, and dynamic pricing models.
5. **Researchers and Policymakers**: The findings can be used as a case study to advocate for the adoption of data-driven approaches in urban planning and policy-making, potentially influencing larger scale initiatives aimed at improving urban mobility and sustainability.

In summary, the project's results offer practical, actionable knowledge that can be leveraged to improve urban transportation efficiency, economic performance of taxi services, and the overall quality of life in urban settings.

**Code & Screenshots**

package IFT512  
  
import org.apache.log4j.{Level, Logger}  
import org.apache.spark.sql.{DataFrame, SaveMode, SparkSession}  
  
object Entrance extends App {  
 Logger.getLogger("org.spark\_project").setLevel(Level.WARN)  
 Logger.getLogger("org.apache").setLevel(Level.WARN)  
 Logger.getLogger("akka").setLevel(Level.WARN)  
 Logger.getLogger("com").setLevel(Level.WARN)  
  
 override def main(args: Array[String]) {  
 val spark = SparkSession  
 .builder()  
 .appName("IFT512-HotspotAnalysis-MYGROUPNAME") // YOU NEED TO CHANGE YOUR GROUP NAME  
 .config("spark.some.config.option", "some-value")//.master("local[\*]")  
 .getOrCreate()  
  
 paramsParser(spark, args)  
  
 }  
   
 private def paramsParser(spark: SparkSession, args: Array[String]): Unit = {  
 var paramOffset = 1  
 var currentQueryParams = ""  
 var currentQueryName = ""  
 var currentQueryIdx = -1  
  
 while (paramOffset <= args.length) {  
 if (paramOffset == args.length || args(paramOffset).toLowerCase.contains("analysis")) {  
 // Turn in the previous query  
 if (currentQueryIdx != -1) queryLoader(spark, currentQueryName, currentQueryParams, args(0) + currentQueryIdx)  
  
 // Start a new query call  
 if (paramOffset == args.length) return  
  
 currentQueryName = args(paramOffset)  
 currentQueryParams = ""  
 currentQueryIdx = currentQueryIdx + 1  
 }  
 else {  
 // Keep appending query parameters  
 currentQueryParams = currentQueryParams + args(paramOffset) + " "  
 }  
 paramOffset = paramOffset + 1  
 }  
 }  
  
 private def queryLoader(spark: SparkSession, queryName: String, queryParams: String, outputPath: String) {  
 val queryParam = queryParams.split(" ")  
 if (queryName.equalsIgnoreCase("hotcellanalysis")) {  
 if (queryParam.length != 1) throw new ArrayIndexOutOfBoundsException("[IFT512] Query " + queryName + " needs 1 parameters but you entered " + queryParam.length)  
 HotcellAnalysis.runHotcellAnalysis(spark, queryParam(0)).limit(50).write.mode(SaveMode.Overwrite).csv(outputPath)  
 }  
 else if (queryName.equalsIgnoreCase("hotzoneanalysis")) {  
 if (queryParam.length != 2) throw new ArrayIndexOutOfBoundsException("[IFT512] Query " + queryName + " needs 2 parameters but you entered " + queryParam.length)  
 HotzoneAnalysis.runHotZoneAnalysis(spark, queryParam(0), queryParam(1)).write.mode(SaveMode.Overwrite).csv(outputPath)  
 }  
 else {  
 throw new NoSuchElementException("[IFT512] The given query name " + queryName + " is wrong. Please check your input.")  
 }  
 }  
}

A computer screen shot of a computer screen

Description automatically generated

package IFT512  
  
import java.sql.Timestamp  
import java.text.SimpleDateFormat  
import java.util.Calendar  
  
object HotcellUtils {  
 val coordinateStep = 0.01  
  
 def CalculateCoordinate(inputString: String, coordinateOffset: Int): Int =  
 {  
 // Configuration variable:  
 // Coordinate step is the size of each cell on x and y  
 var result = 0  
 coordinateOffset match  
 {  
 case 0 => result = Math.floor((inputString.split(",")(0).replace("(","").toDouble/coordinateStep)).toInt  
 case 1 => result = Math.floor(inputString.split(",")(1).replace(")","").toDouble/coordinateStep).toInt  
 // We only consider the data from 2009 to 2012 inclusively, 4 years in total. Week 0 Day 0 is 2009-01-01  
 case 2 => {  
 val timestamp = HotcellUtils.timestampParser(inputString)  
 result = HotcellUtils.dayOfMonth(timestamp) // Assume every month has 31 days  
 }  
 }  
 return result  
 }  
  
 def timestampParser (timestampString: String): Timestamp =  
 {  
 val dateFormat = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss")  
 val parsedDate = dateFormat.parse(timestampString)  
 val timeStamp = new Timestamp(parsedDate.getTime)  
 return timeStamp  
 }  
  
 def dayOfYear (timestamp: Timestamp): Int =  
 {  
 val calendar = Calendar.getInstance  
 calendar.setTimeInMillis(timestamp.getTime)  
 return calendar.get(Calendar.DAY\_OF\_YEAR)  
 }  
  
 def dayOfMonth (timestamp: Timestamp): Int =  
 {  
 val calendar = Calendar.getInstance  
 calendar.setTimeInMillis(timestamp.getTime)  
 return calendar.get(Calendar.DAY\_OF\_MONTH)  
 }  
  
 // YOU NEED TO CHANGE THIS PART  
 def ST\_ZScore(mean: Double, std\_dev: Double, numNeighbors: Int, sigma: Int, numCells: Int): Double =  
 {   
 val numer = sigma - (mean \* numNeighbors)   
 val denom = std\_dev \* Math.sqrt((numCells \* numNeighbors - numNeighbors \* numNeighbors) / (numCells-1))  
 return numer / denom  
 }   
}

A computer screen shot of a black screen

Description automatically generated

package IFT512  
  
import org.apache.log4j.{Level, Logger}  
import org.apache.spark.sql.{DataFrame, SaveMode, SparkSession}  
  
object HotzoneAnalysis {  
  
 Logger.getLogger("org.spark\_project").setLevel(Level.WARN)  
 Logger.getLogger("org.apache").setLevel(Level.WARN)  
 Logger.getLogger("akka").setLevel(Level.WARN)  
 Logger.getLogger("com").setLevel(Level.WARN)  
  
 def runHotZoneAnalysis(spark: SparkSession, pointPath: String, rectanglePath: String): DataFrame = {  
  
 var pointDf = spark.read.format("com.databricks.spark.csv").option("delimiter",";").option("header","false").load(pointPath);  
 pointDf.createOrReplaceTempView("point")  
  
 // Parse point data formats  
 spark.udf.register("trim",(string : String)=>(string.replace("(", "").replace(")", "")))  
 pointDf = spark.sql("select trim(\_c5) as \_c5 from point")  
 pointDf.createOrReplaceTempView("point")  
  
 // Load rectangle data  
 val rectangleDf = spark.read.format("com.databricks.spark.csv").option("delimiter","\t").option("header","false").load(rectanglePath);  
 rectangleDf.createOrReplaceTempView("rectangle")  
  
 // Join two datasets  
 spark.udf.register("ST\_Contains",(queryRectangle:String, pointString:String)=>(HotzoneUtils.ST\_Contains(queryRectangle, pointString)))  
 val joinDf = spark.sql("select rectangle.\_c0 as rectangle, point.\_c5 as point from rectangle,point where ST\_Contains(rectangle.\_c0,point.\_c5)")  
 joinDf.createOrReplaceTempView("joinResult")  
   
 val returnDf = spark.sql("select rectangle, COUNT(point) from joinResult group by rectangle order by rectangle").coalesce(1)  
 return returnDf  
 }  
}

A screenshot of a computer

Description automatically generated

package IFT512  
  
object HotzoneUtils {  
  
 def ST\_Contains(queryRectangle: String, pointString: String ): Boolean = {  
   
 val rectangle\_coordinates = queryRectangle.split(",")  
 val target\_point\_coordinates = pointString.split(",")  
  
 val lat\_point: Double = target\_point\_coordinates(0).trim.toDouble  
 val lon\_point: Double = target\_point\_coordinates(1).trim.toDouble  
 val lat1\_rec: Double = math.min(rectangle\_coordinates(0).trim.toDouble, rectangle\_coordinates(2).trim.toDouble)  
 val lat2\_rec: Double = math.max(rectangle\_coordinates(0).trim.toDouble, rectangle\_coordinates(2).trim.toDouble)  
 val lon1\_rec: Double = math.min(rectangle\_coordinates(1).trim.toDouble, rectangle\_coordinates(3).trim.toDouble)  
 val lon2\_rec: Double = math.max(rectangle\_coordinates(1).trim.toDouble, rectangle\_coordinates(3).trim.toDouble)  
  
 if ((lat\_point >= lat1\_rec) && (lon\_point >= lon1\_rec) && (lat\_point <= lat2\_rec) && (lon\_point <= lon2\_rec)) {  
 return true  
 }  
 return false  
 }  
}

A screenshot of a computer

Description automatically generated

**Creating Jar File:**

sbt assembly

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Running to create output files:**

A computer screen with white text

Description automatically generated

**Output file 1:**

A screenshot of a computer

Description automatically generated

A computer screen with a white line

Description automatically generated

**For output file 2:**

A computer screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

**Output file 2:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Summary & Conclusions**

**Summary**

The "Hotspot Analysis of Geospatial Data" project is likely focused on analyzing taxi trip data to identify areas of high demand, called hotspots, within an urban environment. The project may involve collecting and preprocessing taxi trip data, detecting outliers, and conducting feature engineering to enhance the analysis. Through the use of statistical or machine learning models, the project aims to pinpoint where and when taxis are most frequently requested. The insights garnered from the analysis could be used to optimize taxi dispatching, improve routing, inform urban planning decisions, and enhance overall service efficiency. The project's findings can help taxi companies allocate resources more effectively and potentially offer better service to their customers.

**Conclusion**

The project would provide a comprehensive overview of the patterns and trends observed in urban taxi trip data. The project's analysis likely indicates the most effective methods for distributing taxi services across the city, ensuring that taxis are readily available in high-demand areas and during peak hours. Additionally, it would suggest route optimizations to minimize travel time and maximize operational efficiency.

The project might also highlight potential areas for infrastructural development, such as the expansion of roads or the addition of taxi lanes in chronically congested areas. Policy implications could include recommendations for dynamic pricing during high-demand periods or the establishment of dedicated pick-up spots in busy zones to streamline service.

Furthermore, the findings could underscore the importance of real-time data analysis for rapid response to changing urban dynamics, advising taxi companies on the adoption of smart technologies for live tracking and dispatching. These conclusions would aim to inform taxi companies, urban planners, and policymakers on how to collaboratively enhance the urban transportation network, thereby improving the urban commuting experience for all stakeholders involved.

Top of Form

**References**

1. [ACM SIGSPATIAL GIS Cup 2016](http://sigspatial2016.sigspatial.org/giscup2016/)
2. [Apache Spark™ - Unified Engine for large-scale data analytics](https://spark.apache.org/)
3. Getis Ord Static - [Getis–Ord Spatial Statistics to Identify Hot Spots by Using Incident Management Data - Praprut Songchitruksa, Xiaosi Zeng, 2010 (sagepub.com)](https://journals.sagepub.com/doi/10.3141/2165-05)
4. Zscore - [How to Calculate Z-Score and Its Meaning (investopedia.com)](https://www.investopedia.com/terms/z/zscore.asp)
5. ST\_Contains - [ST\_Contains (postgis.net)](https://postgis.net/docs/ST_Contains.html)