MNNIT Allahabad

DATAVISION

CULRAV AVISHKAR
ANNUAL TECHNO CULTURAL FEST

Manual

Organized by Rasayans, under Avishkar 2024

2024

MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD (INDIA)
सिद्धिर्भवति कर्मजा

Presented By :
Rasayans

MNNIT Allahabad

WELCOME TO
DATA VISION

Welcome to Data Vision, where data meets discovery at Avishkar 2024!

In this data-driven hackathon, prepare to dive deep into the world of data science, wrangle with datasets, and unearth insights that could redefine how we see the world. Here, we challenge you to transform raw data into actionable, meaningful narratives that go beyond the numbers.

Whether you're a seasoned data enthusiast or just starting out, Data Vision is your arena to test your skills, tackle real-world problems, and make an impact with innovative, data-backed solutions. Get ready to analyze, visualize, and optimize—your journey from data to decision-making begins now.

Bring your curiosity, creativity, and code. Let's turn data into a vision for the future!

CODE OF **CONDUCT**

Our hackathon is a space for innovation, collaboration, and learning. To create a positive and productive environment, all participants must adhere to the following guidelines:

Respect and Inclusivity

Treat all participants, mentors, and organizers with respect. Discrimination, harassment, or inappropriate behavior of any kind will not be tolerated. Embrace diversity, and foster a supportive atmosphere for everyone.

Original Work Within Timeframe

All code, designs, and solutions should be created during the hackathon timeframe. Pre-existing projects, third-party code, and templates are allowed only if they are explicitly stated as such and do not constitute the core of the submission.

Collaboration and Fair Play

Work collaboratively, share knowledge, and provide constructive feedback. Refrain from any form of plagiarism or misconduct that undermines fair competition.

Compliance and Accountability

Follow all instructions from organizers, including project submission guidelines, deadlines, and judging criteria. Participants failing to meet these guidelines may be disqualified.

Data Vision
Instructions

Use GitHub

All code must be hosted on GitHub, with clear instructions for setting up and running the application documented in a README.md file. Any environment variables should be securely shared with event coordinators only.

Code of Conduct

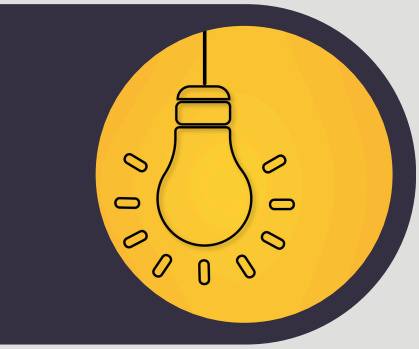
Participants are expected to adhere to the hackathon's Code of Conduct, which fosters respect, collaboration, and inclusivity. Ensure that all team members understand and agree to the guidelines, as failure to comply may result in disqualification.

Deadline and Requirements

Projects must be submitted by the deadline, along with a short presentation (2-5 minutes) showcasing the idea, features, and functionality. Teams should also provide a demo video (up to 3 minutes) or a live demo link if available.

MNNIT Allahabad

Our **Timeline**



Abstract Submission (05/11/2024)

In the initial phase of the hackathon, teams are required to submit an abstract outlining how they are approaching the problem statement. The abstract should include a brief description of the problem you're solving, the technology stack you plan to use, and your proposed solution. This submission helps the judges understand your project's scope and feasibility. Be sure to include any unique or innovative features you intend to implement.

Mid Evaluation (09/11/2024)

At this stage, teams will present their progress to the judging panel. You should demonstrate the core functionality of your project and how you are addressing the problem described in your abstract. The judges will assess your technical implementation, user interface, and how well you're on track to achieve your initial goals. Feedback will be provided to help refine your project before the final submission. Be prepared to show a working demo of your key features.

Final Evaluation (13/11/2024)

For the final evaluation, teams must submit their fully developed projects. This includes a live demo, along with access to your GitHub repository, notebook and other files. Teams will be required to physically present their projects. The judges will evaluate your project on the basis of creativity, technical execution, user experience, and overall impact. Make sure to highlight how your solution effectively solves the problem you set out to address, and demonstrate any advanced features or optimizations you've implemented.

Rules & Eligibility

1. Team/Individual Participation (Max 4 Members)
2. All branches are allowed to participate.
3. Cross-Year Teams are allowed
4. Any tech stack is allowed.
5. **Teams can contact the coordinators for a free temporary OpenAI API Key for implementing GPT models in their project.**

PROBLEM STATEMETS

1. **Urban Traffic Flow Optimization Challenge**
2. **Environmental Sound Classification for Wildlife Conservation**

URBAN TRAFFIC FLOW OPTIMIZATION

With rapid urbanization, cities like Prayagraj are experiencing significant traffic congestion issues. Efficient traffic management is crucial for reducing commute times, lowering emissions, and improving the overall quality of life for residents. You have been provided with a comprehensive dataset containing historical traffic data collected from various intersections across Prayagraj over a week.

Your task is to analyse this dataset to uncover insights into traffic congestion patterns and provide data-driven recommendations to optimize traffic flow in the city. You will use SQL, Excel, or Python for data analysis and showcase your findings through an interactive Tableau or Power BI dashboard.

Dataset Link

URBAN TRAFFIC FLOW OPTIMIZATION

Objective

To accommodate participants from different academic years, the project includes basic tasks tailored for first-year students and medium tasks designed for second and third-year students.

Tasks

(FOR FIRST-YEAR STUDENTS)

1. Data Exploration and Cleaning:

- Import the Dataset:
- Import the "Prayagraj_Traffic_Data.csv" file into your chosen tool (SQL, Excel, or Python).
- Familiarize with the Data:
- Review the dataset to understand each column and its significance.
- Handle Missing Values:
- Identify any missing or null values in the dataset.
- Remove rows with missing data or fill them with appropriate default values.
- Ensure Correct Data Types:
- Verify that each column has the correct data type (e.g., dates, numbers, strings).

2. Descriptive Statistics:

- Calculate Basic Statistics:
- For numerical columns like vehicle_count and average_speed, calculate:

URBAN TRAFFIC FLOW OPTIMIZATION

- Mean
- Median
- Mode
- Minimum and Maximum values
- Summarize Categorical Variables:
- Count the number of occurrences for each category in weather_condition and day_of_week.

3. Basic Data Analysis:

- Identify Peak Traffic Hours:
- Determine the time periods (hours of the day) with the highest vehicle_count for each intersection.
- Intersection Traffic Ranking:
- Calculate the average vehicle_count for each intersection_name.
- Rank the intersections from highest to lowest average traffic.
- Day of Week Analysis:
- Analyse how vehicle_count varies across different days of the week.
- Identify which days have the highest and lowest traffic volumes.

4. Visualization

- Create a Dashboard in Tableau or Power BI:
- **Line Chart** - Plot vehicle_count over time for selected intersections.

URBAN TRAFFIC FLOW OPTIMIZATION

- **Bar Chart -** Show average vehicle_count for each intersection.
- **Pie Chart or Bar Chart -** Display the distribution of traffic across different day_of_week.
- **Optional Map Visualization -** Plot intersections on a map with markers sized based on average vehicle_count.

5. Recommendations:

- Provide Basic Suggestions - Based on your findings, suggest simple measures to improve traffic flow, such as:
- Adjusting signal timings during peak hours.
- Advising commuters about peak traffic times.

(FOR SECOND AND THIRD-YEAR STUDENTS)

1. Advanced Data Cleaning and Preprocessing:

- **Handle Missing Values:**
  - Identify missing or null values in the dataset.
  - Use appropriate imputation techniques (e.g., mean, median) to fill missing numerical values.
  - For categorical variables, consider filling missing values with the mode or a new category (e.g., "Unknown").
- **Detect and Treat Outliers:**

URBAN TRAFFIC FLOW OPTIMIZATION

- Use statistical methods (e.g., Z-score, IQR) to detect outliers in vehicle_count and average_speed.
- Decide whether to remove or adjust outliers and justify your decision.

.**Data Transformation:**

- Extract new features from existing data, such as:
  - hour_of_day from timestamp.
  - is_weekend from day_of_week.
- Normalize or standardize numerical variables if necessary.

2. Intermediate Data Analysis:

- **Weather Impact Analysis:**
  - Analyse how different weather_condition values affect vehicle_count and average_speed.
  - Compare traffic patterns during clear weather versus adverse conditions ('Light Rain', 'Heavy Rain', 'Fog').
- **Holiday and Event Impact:**
  - Examine the effect of holidays (is_holiday = 1) and special events (event includes 'New Year Celebration', 'Local Market Fair') on traffic volumes.
  - Compare average vehicle_count on holidays and special events versus regular days.
- **Correlation Analysis:**
  - Calculate correlation coefficients between variables such as:
  - vehicle_count and average_speed.

URBAN TRAFFIC FLOW OPTIMIZATION

- vehicle_count and pollution_level.
- Identify which factors have the strongest relationships with traffic congestion.

3. SQL Queries or Python Scripting:

Perform the following individual query tasks:

- **Aggregate Data:**
  - Calculate the Total Vehicle Count per Day
    - **Task:** Write a query to calculate the total number of vehicles recorded each day across all intersections.
  - Determine the Average Vehicle Count per Hour Across All Intersections
    - **Task:** Write a query to find the average number of vehicles per hour across all intersections.
- **Filter Data:**
  - Extract Data for Rainy Weather Conditions
    - **Task:** Retrieve all records where the weather_condition is 'Light Rain' or 'Heavy Rain'.
  - Retrieve Weekend Data
    - **Task:** Get all records for which the day_of_week is 'Saturday' or 'Sunday'.
- **Group Data:**
  - Compute Average Speed per Intersection
    - **Task:** Calculate the average average_speed for each intersection_name.
  - Find the Intersection with the Highest Average Vehicle Count
    - **Task:** Identify the intersection with the highest average vehicle_count.

URBAN TRAFFIC FLOW OPTIMIZATION

4. Enhanced Visualization:

- **Develop an Interactive Dashboard:**
  - Heatmap:
    - Task: Create a heatmap showing vehicle_count across different hours and intersections.
  - Interactive Filters:
    - Task: Add filters to your dashboard to allow users to interact with the data.
    - Instructions:
      - Implement filters for:
        - weather_condition
        - day_of_week
        - intersection_name
  - Comparative Charts:
    - Task: Compare traffic patterns during different weather conditions or events.
  - Time-Series Plots:
    - Task: Plot trends over time for variables like vehicle_count, average_speed, and pollution_level.

5. Recommendations:

- **Data-Driven Suggestions:**
  - Task: Based on your analysis, propose strategies to optimize traffic flow.
  - Instructions:
    - Consider factors such as weather conditions, holidays, events, and peak hours.
    - Use evidence from your data analysis to support your recommendations.

URBAN TRAFFIC FLOW OPTIMIZATION

6. Optional Advanced Analysis:

- **Vehicle Type Impact:**
  - Task: Analyse how different vehicle types contribute to overall traffic and pollution levels.
  - Instructions:
    - Examine the counts of cars, buses, trucks, and motorcycles.
    - Assess their impact on pollution_level and congestion.
- **Pollution Analysis:**
  - Task: Study the relationship between traffic volume and pollution_level.
  - Instructions:
    - Calculate correlations or create scatter plots to visualize the relationship.

General Tasks for All Students

1. Documentation:

- **Prepare a Report or Presentation:**
  - Task: Summarize your methodology, findings, and recommendations.
  - Instructions:
    - Include an introduction, analysis sections, conclusions, and recommendations.
    - Use clear language and support your points with visuals from your dashboard.

2. Code Submission:

- **Organize and Document Code:**
  - Task: Submit any SQL queries, Excel workbooks, or Python scripts used.

URBAN TRAFFIC FLOW OPTIMIZATION

- **Instructions:**
  - Ensure your code is well-organized and includes comments explaining each part.
  - Use meaningful variable names and consistent formatting.

Getting Started

1. Access the Dataset:

- Download the "Prayagraj_Traffic_Data.csv" file from the provided link.
- Review the data dictionary to understand each variable.

2. Set Up Your Tools:

- Set up SQL databases if using SQL (e.g., MySQL, PostgreSQL).
- Install Python and libraries like pandas, numpy, matplotlib, seaborn.
- Tableau or Power BI for advanced visualization.

3. Plan Your Work:

- Define the scope of your analysis based on the tasks outlined.
- Allocate time for data cleaning, analysis, visualization, and report preparation.

URBAN TRAFFIC FLOW OPTIMIZATION

Evaluation Criteria

Your submissions will be evaluated based on the following:

1. Data Analysis (40%)

- Accuracy: Correctness of calculations and analysis.
- Depth: Level of insight into traffic patterns and contributing factors.
- Methodology: Appropriate use of statistical methods and tools.

2. Visualization (30%)

- Clarity: Ease of understanding the visualizations.
- Effectiveness: Ability to convey key findings through visuals.
- Interactivity: Inclusion of interactive elements in dashboards.

3. Recommendations (20%)

- Relevance: How well the suggestions address the identified issues.
- Feasibility: Practicality of implementing the recommendations.
- Support: Use of data and analysis to justify recommendations.

4. Communication (10%)

- Presentation: Organization and professionalism of the report or presentation.
- Clarity: Ability to explain your findings in a clear and concise manner.
- Documentation: Quality of code documentation and explanations.

URBAN TRAFFIC FLOW OPTIMIZATION

Final Remarks

This challenge offers a practical opportunity to apply data analytics skills to a real-world problem. Your insights could contribute to improving traffic conditions in urban environments like Prayagraj. We encourage you to approach the task with curiosity and creativity.

Good luck, and we look forward to your valuable contributions!

MNNIT Allahabad

Environmental Sound Classification for Wildlife Conservation

Wildlife conservationists use audio sensors to monitor animal populations and detect illegal activities like poaching. You've been provided with a dataset of audio recordings from a forest reserve, labelled with various animal sounds, ambient noises, and human activities. Your task is to build a system that can automatically classify these sounds to aid in conservation efforts.

DATASET LINK

MNNIT Allahabad

Environmental Sound Classification for Wildlife Conservation

Basic Features

Audio Preprocessing:

- **Feature Extraction:**
  - Convert audio files into feature representations suitable for analysis, such as spectrograms or Mel Frequency Cepstral Coefficients (MFCCs).
- **Normalization:**
  - Normalize audio data to handle variations in recording volumes and lengths, ensuring consistency across the dataset.

Exploratory Audio Analysis:

- **Visualization:**
  - Visualize and compare the audio signatures of different environmental sound classes using plots like spectrograms.
- **Feature Calculation:**
  - Calculate basic audio features such as pitch, tempo, frequency ranges, and duration to understand the characteristics of each class.

MNNIT Allahabad

Environmental Sound Classification for Wildlife Conservation

Medium Features

Sound Classification Model:

- **Model Development:**
  - Build a machine learning model (e.g., Convolutional Neural Network (CNN), Recurrent Neural Network (RNN)) to classify audio files into the 27 predefined classes.
- **Model Evaluation:**
  - Evaluate your model using metrics like accuracy, precision, recall, and F1-score to assess its performance.

Noise Reduction Techniques:

- **Filtering:**
  - Implement noise reduction algorithms to minimize background noise, enhancing the quality of audio inputs.
- **Improvement Analysis:**
  - Analyse how noise reduction impacts classification accuracy and adjust your preprocessing pipeline accordingly.

MNNIT Allahabad

Environmental Sound Classification for Wildlife Conservation

Advanced Features

Real-Time Sound Detection System:

- **Streaming Processing:**
  - Develop a system capable of processing streaming audio data for real-time classification of environmental sounds.
- **Optimization for Edge Devices:**
  - Optimize your model for deployment on edge devices with limited computational resources by techniques like model compression or quantization.

Anomaly and Novelty Detection:

- **Algorithm Implementation:**
  - Implement algorithms to detect unusual sounds or patterns that do not fit into the predefined 27 classes, indicating potential anomalies or new events.
- **Real-Time Alert System:**
  - Propose and develop a method for alerting relevant authorities or systems in real-time upon detecting critical or anomalous sounds.

URBAN TRAFFIC FLOW OPTIMIZATION

Final Remarks

By completing this project, you will contribute to the field of environmental sound analysis, providing tools that can assist in various practical applications such as surveillance, habitat monitoring, and public safety. Your system will leverage the structured dataset to train and validate models that can be deployed in real-world scenarios requiring efficient and accurate sound classification.

Good luck, and we look forward to your valuable contributions!

Thank You

Coordinators

Ayan Basak (+91 98045 45597)
Ashutosh Tewari