**Project Initialization and Planning Phase**

|  |  |
| --- | --- |
| Date | 07 JULY 2024 |
| Team ID | 739702 |
| Project Title | Optimising food delivery |
| Maximum Marks | 3 Marks |

**Project Proposal (Proposed Solution) template**

To explain an Air Quality Index (AQI) analyzer using machine learning (ML), you can structure it similarly to the project proposal template shown in the image. Here's an outline:

|  |  |
| --- | --- |
| **Project Overview** |  |
| Objective | Develop a machine learning system to optimize food delivery processes, minimizing delivery times and enhancing customer satisfaction. |
| Scope | Implement a system that can analyze historical and real-time data to predict optimal delivery routes, estimate delivery times, and allocate resources efficiently. The project will cover data collection, model training, and integration with existing food delivery platforms. |
| **Problem Statement** |  |
| Description | Food delivery services often face challenges in predicting delivery times accurately due to various factors such as traffic conditions, weather, and restaurant preparation times. Inefficiencies in route planning can lead to delayed deliveries, increased costs, and unsatisfied customers. |
| Impact | Optimizing food delivery can significantly enhance customer satisfaction, reduce operational costs, and improve overall efficiency. Accurate predictions and optimized routes can lead to timely deliveries, better resource management, and increased competitiveness in the food delivery market. |
| **Proposed Solution** |  |
| Approach | Utilize supervised machine learning techniques, such as regression models, decision trees, and neural networks, to analyze historical and real-time data for optimizing delivery routes and times. The solution will involve data preprocessing, feature engineering, model training, and evaluation. |

|  |  |  |
| --- | --- | --- |
| **Resource Type** | **Description** | **Specification/Allocation** |
| **Hardware** |  | |
| Computing Resources | High-performance CPUs/GPUs | e.g., 2 x NVIDIA V100 GPUs |
| Memory | Sufficient RAM for large datasets | e.g., 32 GB |
| Storage | Large storage for data, models, and logs. | e.g., 1 TB SSD |
| **Software** |  | |
| Frameworks | Python frameworks | e.g., Tenser flow, sklearn, keras. |
| Libraries | Pandas, NumPy, Matplotlib for data manipulation and  visualization | e.g., numpy, pandas. |
| Development Environment | Jupyter Notebooks, IDEs | e.g., Pycharm |
| **Data** |  | |
| Data | Source: Government and private environmental monitoring agencies, open data | e.g., Kaggle |

|  |  |
| --- | --- |
| Data collection: | Sources: Restaurant data, traffic data, weather data, historical delivery times  Types: Delivery time, traffic conditions, weather conditions, order details CSV, JSON, real-time API feeds |
| Data preprocessing: | Cleaning: Handle missing values, remove outliers  Transformation: Normalize/standardize data  Feature Engineering: Create new features from raw data |
| Model Training: | Algorithms: Linear regression, random forest, gradient boosting, deep learning models  Evaluation: Cross-validation, performance metrics (RMSE, MAE, R2 score)  Integration: Real-time data ingestion and prediction  Visualization: Dashboards and alerts for delivery times and route optimization |

|  |  |  |
| --- | --- | --- |
|  | platforms (e.g., Kaggle) Size: Varies depending on the region and time span Format: CSV, JSON, real-time API feeds |  |

**Resource Requirements**