1. **Statement - Item Categorization. 40 marks**

Description: Create a model or research the necessary steps to create a model for categorizing items. When the cook adds an item to their kitchen, it should be automatically categorized into multiple categories. We can provide the sample data for this to train the model.

For instance:

- Idly - South Indian, Protein Rich, Breakfast, Baked Items etc.
- Chicken Vindaloo - North India, Punjabi, Non-Veg, Chicken, Protein Rich etc.
- Ragi Dosa - South Indian, Diabetic Friendly, Millet Based, Pregnancy friendly etc.

2. **Statement - Last Mile Delivery Batching. 60 marks**

Description: It is crucial in today's last-mile delivery ecosystem to optimize for speed, and cost efficiencies. Smarter algorithms play a crucial role in the ecommerce marketplace deliveries We need to group/batch the delivery of multiple items to the same rider without losing time. Here are several scenarios where we need smart operational research algorithms.

Rule # 1:

- Two orders - From the same kitchen.
- To the same customer.
- Ready at the same time (10 mins apart).
- Assign the pick-up to the same rider.

Rule # 2:

- Two orders.
- From two different kitchens (1 km apart).
- To the same customer.
- Ready at the same time (10 mins apart).
- Assign the pick-up to the same rider.

Rule # 3:

- Two orders.
- From the same kitchen.
- To two different customers (1 km apart).
- Ready at the same time (10 mins apart).
- Assign the pick-up to the same rider.

Rule # 4:

- Two orders.
- From two different kitchens (1 km apart).
- To the same customer.
- Ready at the same time (10 mins apart).
- Assign the pick-up to the same rider.

Rule # 5:

- Two orders.
- From two different kitchens (1 km apart).
- To the same customer.
- Ready at the same time (10 mins apart).

- Assign the pick-up to the same rider.

Rule # 6:

- Two orders.
- To the same customer.
- 2nd kitchens pick up on the way to the customer.
- Ready at the time the rider reaches the second kitchen (10 mins apart).
- Assign the pick-up to the same rider.

Rule # 7:

- Two orders.
- 2nd customers drop on the way to the 1st customer (Vice Versa).
- 2nd kitchens pick up on the way to the customer.
- Ready at the same time (10 mins apart or by the time rider reaches the kitchen).
- Assign the pick-up to the same rider.

Rule # 8:

- Two orders.
- From the same kitchen.
- 2nd customers drop on the way to the customer 1st (Vice Versa).
- Ready at the same time (10 mins apart).
- Assign the pick-up to the same rider.

3. **Statement - Predictive Maintenance for Infrastructure. 60 marks**

Description: Create AI models or research the necessary steps to create AI models that can predict potential failures or issues in infrastructure components (e.g., servers, networks) based on historical data. This could help DevOps teams proactively address issues before they cause downtime or performance degradation.

For instance:

AI models are trained on historical data regarding cluster performance, node utilization, and workload patterns. These models analyse metrics such as CPU usage, memory consumption, and network traffic to predict potential failures or performance degradation. For instance, if the model detects a gradual increase in CPU usage on certain nodes, it may predict an impending failure due to resource exhaustion. The DevOps team receives automated alerts, allowing them to proactively address the issue by scaling up resources or reallocating workloads before any downtime occurs. This predictive maintenance approach helps optimize the reliability and stability of the EKS environment, ensuring continuous availability of applications running on Kubernetes.

4. **Statement - Automated Code Reviews and Quality Assurance. 40 marks**

Description: Develop AI-powered tools to analyse code changes and automatically identify potential bugs, security vulnerabilities, or coding best practice violations. This could help streamline the code review process and improve overall code quality.

For instance:

AI-powered code review tool that automatically scans code changes for bugs, security flaws, and coding best practices. This tool provides instant feedback to developers, streamlining the review process and enhancing overall code quality. By leveraging this tool, the team ensures that only high-quality code is merged into the project, reducing the risk of bugs and vulnerabilities. Additionally, the tool learns from past reviews, continuously improving its analysis and further enhancing code quality over time.