Answer 1: Business KPI

1.Number of the dents/Scratches

Calculation: Count of the dents/scratches

What does it measure: Measures the number of dents/scratches in used retails vehicles.

2.Area of the dent/length of the scratch

Calculation: Length of the dent * breadth of the dent

What does it measure: Measures the Area of damage of the used retails vehicles.

3.Retail Gross per Unit

Calculation: Total Used Retail Gross Profit ÷ No. of Used Retail Units Sold

What does it measure: How much gross profit was earnt on average for each used car retailed.

4.Wholesale Gross per Unit

Calculation: Total Used Wholesale Gross Profit ÷ No. of Used Wholesale Units Sold

What does it measure: How much gross profit was earnt on average for each used car wholesaled.

Why is it important: Low to negative figures for this KPI may indicate that a dealership is not getting any particular benefit from wholesaling or trade-ins. The used cars the dealership is accepting may not have many prospects.

5.Net Gross per Used Vehicle Retailed

Calculation: (Used Retail Gross Profit + Used Wholesale Gross Profit) ÷ No. of Used Retail Units Sold

What does it measure: Vehicle gross profit generated by the used vehicle department expressed on a per retail unit basis.

6.Units Sold per Sales Employee per Month

Calculation: No. of Retail Units Sold ÷ No. of Used Vehicle Sales Employees

What does it measure: The average number of retail used vehicles sold by each member of the used vehicle sales team.

Why is it important: A measure of salesperson's ability to sell vehicles and indicates the level of throughput at the dealership.

7.Gross per Sales Employee per Month

Calculation: Used Vehicle Gross Profit[1] ÷ No. of Used Vehicle Sales Employees

What does it measure: The average gross profit generated by each member of the used vehicle sales team.

Why is it important: A measure of salesperson's ability to be profitable when making vehicle sales.

8.Total Selling Expenses % Gross

Calculation: (Used Vehicle Department Variable Expenses + Used Vehicle Department Semi-Fixed Expenses) ÷ Used Vehicle Gross Profit[1]

What does it measure: How much it costs to sell used vehicles as a proportion of how much profit the used vehicle department makes on the same sales. Note, this KPI does not reflect overheads but only those expenses that are considered directly related to the sale of vehicles i.e. 'selling expenses

Why is it important: Observing this total figure indicates if the costs are too high in the department given the level of sales at the dealership. Drilling down into the particular expense categories enables effective expense management and control.

9.Used Vehicle Selling Gross % Gross profit

Calculation: (Used Vehicle Gross Profit[1] - Used Vehicle Variable Expenses - Used Vehicle Semi Fixed Expenses) ÷ Used Vehicle Gross Profit

What does it measure: How much gross profit is actually retained by the used vehicle department.

Why is it important: Selling gross % gross profit is the measure of departmental profitability in Profit reports. Even if a dealership manages a high level of sales volume, a low selling gross % gross figure indicates that the used vehicle department is still not that profitable. Often this may require a dealership to investigate the expenses of the department or the grosses required to sustain the current level of expenses.

10.Selling Gross per Used Vehicle Retailed ($)

Calculation: (Used Vehicle Gross Profit[1] - Used Vehicle Variable Expenses - Used Vehicle Semi Fixed Expenses) ÷ No. of Used Retail Units Sold

What does it measure: How much gross profit was retained on average for each used vehicle sold by the dealership.

Why is it important: It is useful to compare this KPI to Gross Profit per Used Vehicle Sold. If this figure is low, but 'gross per unit' is high - the dealership should investigate the department's expense control.

11.Selling Gross per Used Vehicle Sales Employee

Calculation: (Used Vehicle Gross Profit[1] - Used Vehicle Variable Expenses - Used Vehicle Semi Fixed Expenses) ÷ No. of Used Vehicle Sales Employees

What does it measure: How much gross profit was retained on average by each used vehicle sales employee.

Why is it important: It is useful to compare this KPI to Gross per Sales Employee per Month. If this figure is low, but 'gross per sales employee' is high - the dealership should investigate the department's expense control.

12.Used Vehicle Days' Supply

Calculation: (No. of Used Vehicles on Hand ÷ No. of Retail Used Vehicles Sold) × No. of Days in Month i.e. 30.4 days

What does it measure: How many days the dealership can sustain sales for based on levels of used vehicle stock and current sales performance.

Why is it important: Indicates how well used vehicle stock levels are controlled. Consistently high days supply figures should be addressed as there is a direct correlation to stock holding costs of the dealership e.g. floorplan interest.

13.Ageing of Stock

Calculation: Value of Stock Held for 0-30* Days ÷ Value of Total Stock on Hand
*Ageing categories are 0-30 days, 31-60 days, 61-90 days, 90+ days.

What does it measure: Measures the mix of stock based on the date they were brought into stock.

Why is it important: Indicates how long dealers are holding their stock (i.e. being unable to sell them).

14.Average Cost on Hand

Calculation: Value of Used Vehicle Stock on Hand ÷ No. of Used Vehicles on Hand

What does it measure: The average cost to holding a used vehicle at the dealership.

Why is it important: Provides an understanding of the value of the used vehicles the dealership has available to sell.

15.Average Cost of Sale

Calculation: (Used Vehicle Retail & Wholesale Sales - Used Vehicle Retail & Wholesale Gross Profit) ÷ No. of Retail Used Vehicles Sold

What does it measure: The average cost price of the retail vehicles that the dealership sells.

Why is it important: It is useful to compare the average cost that a dealership sells its used vehicles at to the cost to the dealership of holding the cars itself. If the average cost of sale is significantly lower than the average cost on hand, this may reflect a lost gross profit opportunity.

16.ROI (Gross ROI)

Calculation: (Used Vehicle Gross Profit[1] ÷ Used Vehicle Cost of Sales) × (365 Days ÷ Used Vehicle Days Supply)

What does it measure: A measure of the efficient utilisation of current investment in inventory. The higher the figure for the KPI, the better return a dealership is getting from holding stock.

Answer 2: Need and advantage of the MLOPS solution.

Collaboration

When we look at the traditional setting of all teams working on an ML project, the software development, data science, and IT operations teams, we see an uncorrelated and isolated group of individuals with different expertise. MLOps brings a more collaborative process that makes all teams work hand-in-hand to combine their expertise in building more efficient, fast, and scalable machine learning models that leverage all fields.

Automation

The benefits of automation in software and machine learning development are very crucial to achieve desired business results. Rather than wasting time repeating the same processes involved in the ML-powered software's lifecycle, automation also helps various teams focus on more critical business issues to drive more fast and reliable business solutions.

Rapid Innovation

Improved collaboration and automation in machine learning by MLOps fosters rapid innovation and development of new feature-rich machine learning solutions.

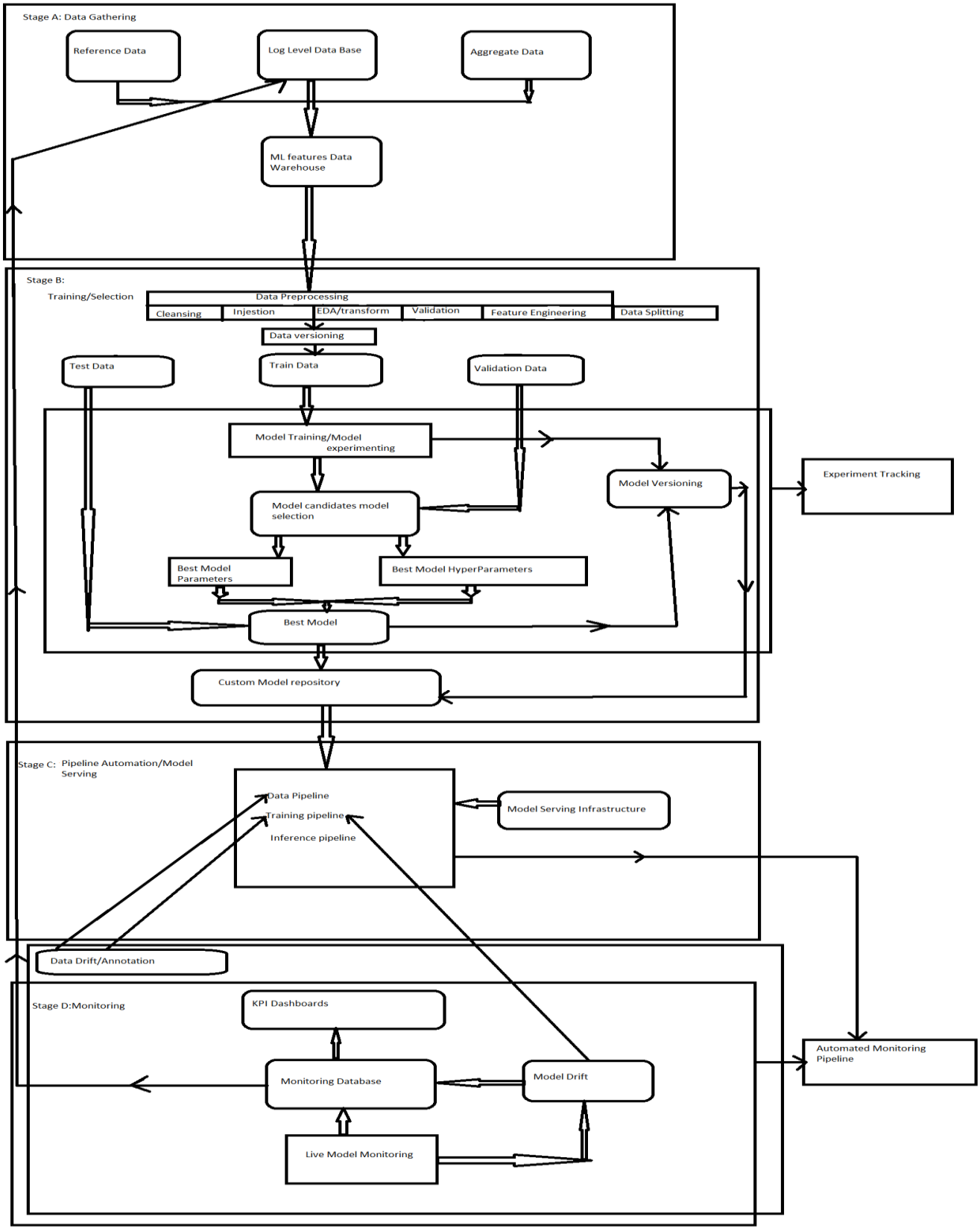
Easy Deployment

MLOps makes it easy to deploy a machine learning algorithm through a streamlined process like that in DevOps. It improves the continuous integration, deployment, and delivery of machine learning models.

Effective Lifecycle Management

MLOps ultimately improves all production teams' efficiency and the entire machine learning project development workflow right from design to deployment.

Answer3: Architecture



Answer 4: Reason for the Tool used.

Docker:

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

Docker provides the ability to package and run an application in a loosely isolated environment called a container. The isolation and security allows you to run many containers simultaneously on a given host. Containers are lightweight and contain everything needed to run the application, so you do not need to rely on what is currently installed on the host. You can easily share containers while you work, and be sure that everyone you share with gets the same container that works in the same way.

Docker provides tooling and a platform to manage the lifecycle of your containers:

- Develop your application and its supporting components using containers.
- The container becomes the unit for distributing and testing your application.
- When you're ready, deploy your application into your production environment, as a container or an orchestrated service. This works the same whether your production environment is a local data centre, a cloud provider, or a hybrid of the two.

Model registry:

A model registry is a repository used to store and version trained machine learning (ML) models. Model registries greatly simplify the task of tracking models as they move through the ML lifecycle, from training to production deployments and ultimately retirement.

In addition to the models themselves, a model registry stores information (metadata) about the data and training jobs used to create the model. Tracking these requisite inputs is essential to establish lineage for ML models. In this way, a model registry serves a function analogous to

version control systems (e.g. Git, SVN) and artifact repositories (e.g. Artifactory,) for traditional software.

GIT:

Git is a distributed version control system, which means that a local clone of the project is a complete version control repository. These fully functional local repositories make it easy to work offline or remotely. Developers commit their work locally, and then sync their copy of the repository with the copy on the server. This paradigm differs from centralized version control where clients must synchronize code with a server before creating new versions of code.

Git's flexibility and popularity make it a great choice for any team. Many developers and college graduates already know how to use Git. Git's user community has created resources to train developers and Git's popularity make it easy to get help when needed. Nearly every development environment has Git support and Git command line tools implemented on every major operating system.

Kubernetes:

Keeping containerized apps up and running can be complex because they often involve many containers deployed across different machines. Kubernetes provides a way to schedule and deploy those containers—plus scale them to your desired state and manage their lifecycles. Use Kubernetes to implement your container-based applications in a portable, scalable, and extensible way.

Answer5: Workflow

Concept drift

Concept drift happens when the relationship between input variables and the target variable changes. This means that the definition of what we are trying to predict changes so that our model provides inaccurate predictions. This change can be gradual, sudden, or recurring.

- **Gradual concept drift**: The change in fraudulent behaviour is an example of gradual concept drift. As fraud detection methods become more sophisticated, fraudsters adapt to evade fraud detection systems by developing new strategies. An ML model trained on historical fraudulent transaction data would be unable to classify a new strategy as fraud. This means that the performance of the model would degrade because what is classified as fraud has changed over time.
- **Sudden concept drift**: The Covid-19 pandemic suddenly changed consumer behaviour. For instance, consumer spending on recreational durable goods such as home fitness equipment increased by 18%, while spending on transportation services decreased by 23% in 2020. A demand forecasting model trained with pre-pandemic data would not predict these changes in consumer habits.
- **Recurring concept drift**: This is also called seasonality. For instance, retail sales increase significantly during the Christmas season or on Black Friday. An ML model that does not take these known recurring trend changes into account would provide inaccurate predictions for these periods.

Data drift

Data drift occurs when the properties of the input data change. For instance, as an online platform grows, the age distribution of its users may change over time. Since the usage habits of young and old people are not the same, a model trained on young people's usage data would provide inaccurate predictions for old people's behaviour.

How to deal with model drift?

1.Monitor the performance of the model

In order to deal with model drift, you should first be able to detect it before it causes major problems for end-users. Determining model performance metrics and continuously monitoring the performance of your model against them is therefore key to the long-term success of ML models.

2.Check data quality

Some rapid performance changes can be due to problems in training data quality such as biases in data rather than concept or data drift. If that is the case, the problem would reveal itself early when you apply your model in a real-world use case.
As an example, Google Health developed a deep learning model to detect a retina disease from patients' eye scans. The model had 90% accuracy during its training phase but it failed to provide accurate results in real-life. This is because the model is trained with high-quality eye scans while real-world eye scans were lower in quality.

Ignoring a known seasonality is also a data quality issue. If your training data does not include recurring changes in the data, such as soaring retail sales during the Christmas season, this is a data quality issue that can be easily fixed.

3.Retrain the model

If you detect a concept or data drift, you can retrain your model with more recent data. Depending on the nature of the drift, there are different approaches:

- Use only recent data if old data has become outdated,
- Use all available data if the old data wouldn't cause inaccurate predictions,
- If the deployed model allows weighting, use all available data but assign higher weights to recent data so that the model pays less attention to old data.

Another option is online learning where the model continuously learns in real-time with the data feed. This will enable the model to keep itself up to date with evolving datasets.

4.Tune the model

If retraining the model doesn't suffice, rebuilding the model can also help. This is because you have built your model with old training data in mind. Running multiple experiments with different features, hyperparameters, model architectures, etc. can help you update your model to keep in line with new data.

1.What component/pipeline will be triggered if there is any drift detected?What if the drift detected is beyond an acceptable threshold?

- If data drift is detected then usually training pipeline is detected given the performance of the model is satisfactory.
- If the data drift is detected and model perfomance is not satisfactory then the data pipeline will be triggered first and then the training pipeline.
- In above mention both cases, retraining of the model will occur.
- If there is a concept drift then training pipeline will be triggered to retrain the model.

2.What component/pipeline will be triggered if you have additional annotated data?

- If you have additional annotated data, then the data pipeline will be triggered then training pipeline will be triggered to retrain the model.

Workflow Steps:

Workflow has been divided into upper layer:pipelines and lower layer:drives.

Upper Layer: Pipelines

- **Data and Model Experimentation**
    1. Data gathering
    2. Data preprocessing
    3. Build data base
    4. Load data
    5. Model training/experimenting
    6. Experiment tracking

- **Automation of data pipeline**
    1. Check_raw date schema
    2. Load data
    3. Data preprocessing
    4. Check model input schema

- **Automation of training pipeline**
    1. Data preprocessing
    2. Train model
    3. Track model
    4. Update model version
- **Automation of inference pipeline**
    1. Data preprocessing
    2. Check input features
    3. Generating model predictions
- **Continous monitoring pipeline**
    1. Get data drift
    2. Get concept drift
    3. Get data annotation
    4. Train model

Lower Layer: Drives

- **Data-** training data, test data and monitoring data.
- **Code:** training code, test code and Application code.
- **Artifacts:** Trained model, packaged model and production model.
- **Middleware:** GIT, model registry, dockers and kubernetes.
- **Infrastructure:** training computer,Production compute, central storage and feature store.

Workflow Diagram