Problem statement

Creating an ML system that has the features of a complete production stack, from experiment tracking to automated model deployment and monitoring. The framework should also be able to train if additional annotations are available later.

among the various types of damage, the model should be able to detect two: scratches and dents. It should return “None” if no damage is detected.

##--##

Q1. System design: Based on the above information, describe the KPI that the business should track.

- Minimizing the heavily damaged vehicles into inventory stock by identifying damages /scratches and reducing manual intervention and reducing human resource required to validate and increase in Number of sales

##--##

Q2. System design: Your company has decided to build an ML-Ops system. What advantages would you get from building an ML-Ops system rather than a simple model?

- ML-OPS is the best and proven way to deploy models into production for prediction and maintenance
- it reduces the human effort required to maintain models manually.
- ML-Ops helps in automating end to end solution and reduces over the box efforts on developers.
- it helps in continuous deployment/training from identifying data drift /triggering /Testing/inference all tasks could be automated – which sets the bench mark in deploying and maintaining ML models.

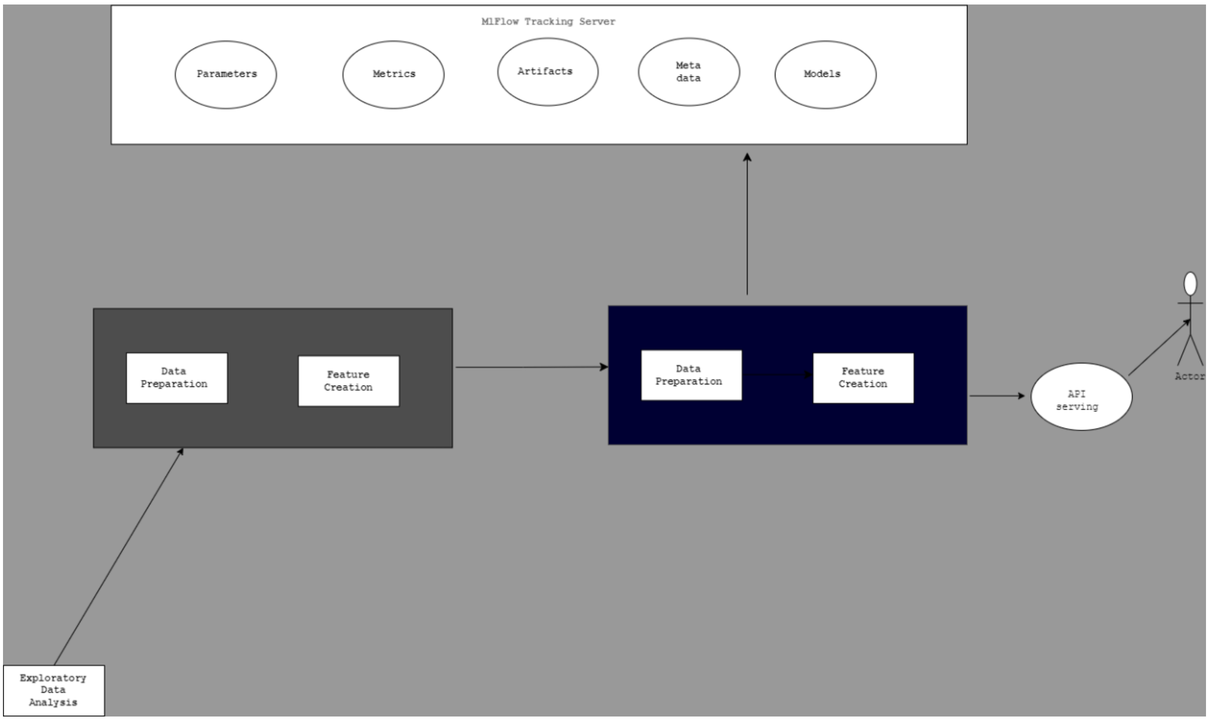
##--##

Q3. **System design:** You must create an ML system that has the features of a complete production stack, from experiment tracking to automated model deployment and monitoring. For this problem, create an ML system design (diagram)

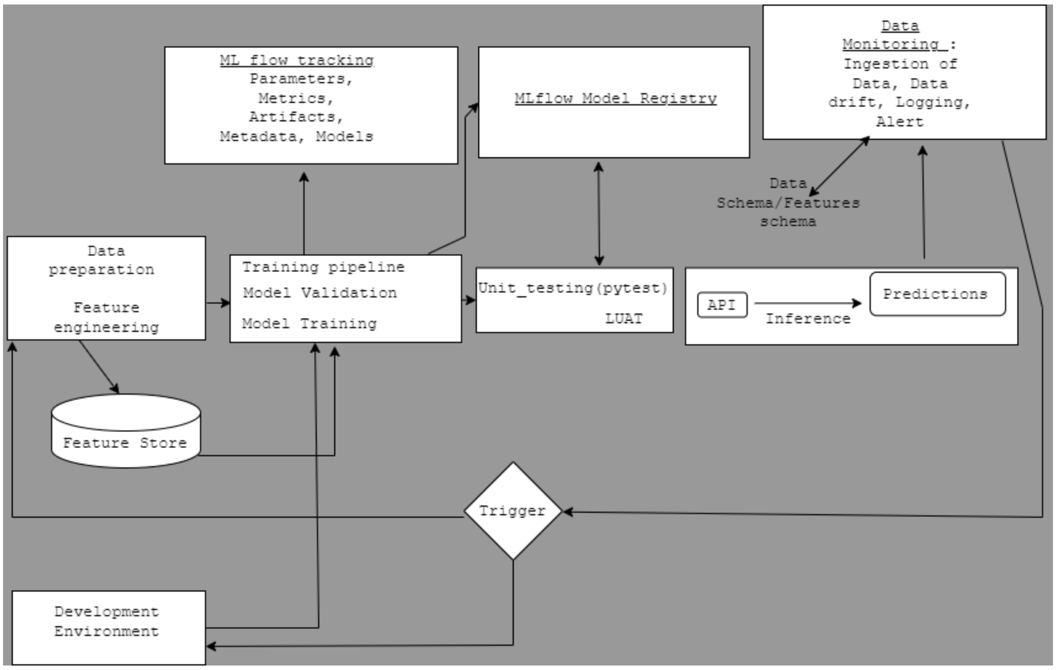
while designing an MLOps-based solution for a problem, we must understand the root problem and design its solution iteratively. Here the system architecture is divided into two environments.

Development environment.

- Exploratory data analysis
- Data preparation
- Model training and tuning- Rapid experimentation is essential to selecting the baseline model in this stage. Here, we can experiment with multiple models/algorithms, PyCaret library can be used as an approach by reducing the number of lines of code, libraries, and tasks required to build models, which allows experimentation using various models.
- Model validation



- Model serving (Production environment): environment is where you operationalise the best model identified from the development environment/stage.
  - Ml-flow to manage the ML lifecycle, including experimentation, reproducibility, deployment, and a central model registry. it is an excellent solution for tracking all your model development experiments.
  - Apache Airflow, an open-source orchestration tool widely adopted by the industry. building complex pipelines in the form of directed acyclic graphs using which you can, create manage pipeline Manage pipeline Monitor pipeline.
  - Evidently: open-source tool, which helps in evaluating and monitoring models in production



##---##

Q4. **System design:**

ML-OPS steps and tools used:

- The ML system depends upon the use case you are working with and thus varies along the business needs and infrastructure.
  - Data and model experimentation- (Pandas profiling, Pycaret, Mlflow)
  - Automation of data/training/inference pipeline —(Airflow)
  - Continuous Serving/monitoring pipeline- (Evidently & Streamlit)

ydata-profiling - pandas-profiling package naming was changed. It offers report generation for the dataset with lots of features and customizations for the report generated.

Pandas Profiling also helps a lot in Exploratory Data Analysis (EDA). EDA is used to understand the underlying structure of data, detect patterns, and generate insights in a visual format.

For EDA, we have to write many lines of code, which can sometimes be complex and time-consuming, but it can be automated using Pandas Profiling with just a few lines of code.

- Ease of use: Pandas profiling is very easy to use, only need to write a couple of lines of code to generate a comprehensive report.
- Time-saving: Pandas profiling can create a comprehensive report with a wide range of information about a dataset with minimal effort, makes it a great option for EDA.
  - Interactive HTML reports: Pandas profiling generates interactive HTML reports that are easy to analyze and understand. The reports also allow you to dig deeper into specific variables and explore their distributions.

    Reference: https://docs.profiling.ydata.ai/latest/

Pycaret - PyCaret is an open-source, low-code machine learning library in Python that automates machine learning workflows. It is an end-to-end machine learning and model management tool that exponentially speeds up the experiment cycle and makes you more productive.

- PyCaret is an alternate low-code library that can be used to replace hundreds of lines of code with a few lines only. This makes experiments exponentially fast and efficient.
- PyCaret is essentially a Python wrapper around several machine learning libraries and frameworks, such as scikit-learn, XGBoost, LightGBM, CatBoost, spaCy, Optuna, Hyperopt, Ray, and a few more.

  Reference: https://pycaret.org/

Mlflow – MLflow is an open-source platform for managing the end-to-end machine learning lifecycle. It has the following primary components:

- Tracking: Allows you to track experiments to record and compare parameters and results.
- Models: Allow you to manage and deploy models from a variety of ML libraries to a variety of model serving and inference platforms.

- Projects: Allow you to package ML code in a reusable, reproducible form to share with other data scientists or transfer to production.

- Model Registry: Allows you to centralize a model store for managing models' full lifecycle stage transitions: from staging to production, with capabilities for versioning and annotating. Databricks provides a managed version of the Model Registry in Unity Catalog.

- Model Serving: Allows you to host MLflow models as REST endpoints. Databricks provides a unified interface to deploy, govern, and query your served AI models.

  Reference: https://mlflow.org/

<u>Airflow</u>: is a platform to programmatically author, schedule, and monitor workflows.

- When workflows are defined as code, they become more maintainable, versionable, testable, and collaborative. Use Airflow to author workflows as directed acyclic graphs (DAGs) of tasks.

- The Airflow scheduler executes your tasks on an array of workers while following the specified dependencies. Rich command line utilities make performing complex surgeries on DAGs a snap.

- The rich user interface makes it easy to visualize pipelines running in production, monitor progress, and troubleshoot issues when needed.

  Reference: https://airflow.apache.org/

<u>Evidently</u>: Evidently is an open-source Python library for data scientists and ML engineers.

It helps evaluate, test, and monitor data and ML models from validation to production. It works with tabular, text data and embeddings.

<u>Data drift algorithm</u>

Evidently uses the default Data Drift Detection algorithm. It helps detect the distribution drift in the individual features, prediction, or target.

Evidently compares the distributions of the values in a given column (or columns) of the two datasets. You should pass these datasets as **reference** and **current**. Evidently applies several statistical tests and drift detection methods to detect if the distribution has changed significantly. It returns a "drift detected" or "not detected" result.

Reference: https://docs.evidentlyai.com/

<u>Streamlit</u> -Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science.

Reference: https://docs.streamlit.io/

##--##

Q5. Workflow of the solution:

- The detection network can either have all the damages or we can have different networks trained for specific groups of damages.
- In this scenario ill choose the most effective approach is the different network for different damages. The reason behind this is that different damages have different characteristics.
- Reason for not choosing a single model trained for all damages because it msy decrease the performance of the model.

Common strategy of object detection in CV is using algorithms of deep learning – examples are Faster-RCNN, Retinanet, YOLO, etc. These deep learning networks can be modified by changing the backbone used to extract

Damage Identification by Object Detection

Damages like a tear on the metal body and dislocation or gaps between parts are easily identifiable using detection networks. They give accurate results when trained on large and represented data.

Scratch Detection

Scratches on the metal body and dislocation are bit difficult using detection networks. They give false positive results. suggestion is to use an ensemble of detection as well as segmentation techniques. prediction generated using a segmentation model overlaid onto the original image

Metrics: for any ML or DL model, to determine how well our model performs on the test data. These metrics can be F1-score.

To get values of precision and recall, which in turn depends on IOU.

IOU= area of overlap/area of union

IOU is a metric that computes the ratio of intersection between the ground truth box and predicted box of the model to area of union of two boxes. The value varies between 0 and 1. It tells how well boxes have overlapped.

ML-OPS steps and tools used:

- The ML system depends upon the use case you are working with and thus varies along the business needs and infrastructure.
  - Data and model experimentation- (Pandas profiling, Pycaret, Mlflow)
  - Automation of data/training/inference pipeline —(Airflow)
  - Continuous Serving/monitoring pipeline- (Evidently & Streamlit)

Step1: Data cleaning and EDA

Step2: Model experimentation and tracking using Pycaret & Ml-OPS in Development environment.

Step3: setup Ml-flow

Step 4: Setup Airflow

Step 5: Choose the best model & Hypertune it

Step 6: map trigger on Airflow for Data inference and training

Step 7: setup & map ML-flow for production environment

Step 8: Create & run- Continuous monitoring pipeline- (Streamlit & Evidently- generating the data drift report & Actions to be taken, ensuring the new data you are getting is in the correct format- If there is any deviation between the two schemas/ distribution, an alert is raised to the respective stakeholders to detect data drift.)

Conclusion

Automated car damage detection reduces human error to considerable level. Moreover, vehicle inspection technology improves over time as more data helps train the algorithms. As mentioned in above workflow section, ensemble techniques further enhance the accuracy and reliability of vehicle inspection automation.

Reference: https://inspektlabs.com/blog