1. What is DVC, and why is DVC used?

DVC is a system for data version control. It is essentially like Git but is used for data. With DVC, you can keep the information about different versions of your data in Git while storing your original data somewhere else.

2. How is DVC different from git and GitHub?

Git is a open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency; While DVC is designed to handle large files, data sets, machine learning models, and metrics as well as code. DVC helps machine learning teams manage large datasets, make projects reproducible, and better collaborate.

3. Which command can be used to initialize a DVC project?

Initialize a DVC project by typing following command.

dvc init

4. In What all use cases DVC can be used?

Uses of DVC can be mainly spit into two categories to cover all features easily - Data Management as well as Experiment Management.

**Data Management**

* **Data and model versioning** is the base layer of DVC for large files, datasets, and machine learning models. Use a standard Git workflow, but without storing large files in the repo. Data is cached by DVC, allowing for efficient sharing. Think "Git for data".
* **Data pipelines** describe how models and other data artifacts are built, and provide an efficient way to reproduce them. Think "Makefiles for data and ML projects" done right.
* **Data and model acces**s shows how to bring, explore, and access data artifacts from outside the project. This can help download a specific version of an ML model to a deployment server or import a dataset into another project, for example.
* **Metrics, parameters, and plots** can be attached to pipelines. These let you capture, evaluate, and visualize ML projects without leaving Git.

**Experiment Management**

* **Experiments** enable exploration, iteration, and comparison across many ML experiments. Track your experiments with automatic versioning and checkpoint logging. Compare differences in parameters, metrics, code, and data. Apply, drop, roll back, resume, or share any experiment.
* **Visualization** helps you compare experiment results visually, track your plots, and generate them with library integrations.

5. Which command can be used to reproduce the entire pipeline?

*dvc repr*

This command will reproduce complete or partial pipelines by executing commands defined in their stages in the correct order.

6. Which DVC command can be used to check metrics?

*dvc metrics :-*  This command can we used with positional arguments

*dvc metrics show :-* Print metrics, with optional formatting

*dvc metrics diff :-* Show changes in metrics between commits.

7. Can we store a large amount of Data on GitHub? Justify.

Git is a source code versioning system, designed for use with large collections of small text files, generally no more than a few kilobytes in size.GitHub blocks files larger than 100 MB. To track files beyond this limit, you must use Git Large File Storage (Git LFS).