* **Jupyter Notebook** - An open source web application to create and share documents containing live code, visualizations, and text.
* **Colab Notebook** - Jupyter notebooks hosted by Google that provide free access to GPUs and machine learning tools.
* **GitHub** - A Git repository hosting service used to store and manage code as well as track changes. Enables collaboration.
* **Virtual Environment** - An isolated Python environment that allows packages to be installed for use in a particular application rather than globally.
* **README File** - A text file that introduces and explains a project. It contains info to help others understand and contribute.
* **Requirements File** - A text file listing all the Python package dependencies needed to run an application. Allows reproducible builds.
* **Makefile** - A file containing a set of directives to automate building, testing, and managing a project.
* **Continuous Integration** - The practice of frequently merging code changes and automatically building and testing code to catch issues fast.

**Summary of Lesson** This lesson provided an overview of key tools and best practices for getting started with data science, including using Jupyter Notebooks, Colab, GitHub for sharing projects, structuring reproducible notebooks, leveraging makefiles and requirements for build automation, and continuous integration workflows.

**Top 3 Key Points :-**

* + - 1. Jupyter Notebooks enable iterative data science experiments
      2. GitHub facilitates collaboration on data science portfolio projects
      3. Makefiles, requirements, and CI improve reproducibility
* **Heuristic** - A practical problem-solving approach not guaranteed to be optimal but sufficient for reaching an immediate goal.
* **Greedy Algorithm** - An algorithm that follows the heuristic of making locally optimal choices at each stage in hope of finding a global optimum.
* **Traveling Salesman Problem (TSP)** - A classic computer science optimization problem involving finding the shortest route visiting each city in a list exactly once.
* **Simulation** - Imitating a real-world process over time with a mathematical model to study system behavior and performance.
* **Law of Large Numbers** - The principle that the larger the sample size in a probabilistic simulation, the more it reveals the true underlying statistical distributions.
* **Experiment Tracking** - The MLOps practice of tracking key metrics like performance across iterations of an machine learning experiment.

**Summary of Lesson** This lesson explained common heuristics like greedy algorithms and using simulations to discover optimized solutions, comparing them to ML experiment tracking.

**Top 3 Key Points**

1. Heuristics provide practical approaches to complex optimization problems
2. Simulations imitate real-world systems to study behaviors
3. MLOps experiments also iterate to find optimal models

* **Supervised Learning** - A type of machine learning using labeled historical data to train models to predict unlabeled future data.
* **Unsupervised Learning** - A machine learning approach that finds hidden patterns and relationships in unlabeled data.
* **Clustering** - An unsupervised learning technique that groups data points based on similarity, assigning them to clusters.
* **K-Means Clustering** - A popular clustering algorithm that partitions data into k clusters by computing the mean of all data points assigned to a cluster.
* **Diagnostics** - Visual tools like elbow plots, silhouette analysis, and distance maps used to evaluate and tune the performance of clustering models.
* **Parallelization** - Running clustering computations like k-means concurrently across multiple CPU cores to improve performance

**Summary of Lesson** This lesson explained clustering for unsupervised learning, specifics of k-means clustering, using diagnostics to evaluate models, and leveraging parallel processing to scale expensive k-means computations.

**Top 3 Key Points**

* Clustering finds natural data groupings without labels
* K-means forms clusters based on centroids and distance functions
* Diagnostic plots and parallelization optimize k-means