**1. Databases & SQL**

* **ACID Properties**: Understand atomicity, consistency, isolation, and durability.
* **Normalization**: Master the concepts of database normalization (1NF to 3NF).
* **Constraints**: Primary key, foreign key, unique, and check constraints.
* **Relationships**: One-to-one, one-to-many, many-to-many.
* **Joins**: INNER, LEFT, RIGHT, and FULL OUTER JOIN.
* **3-Schema Architecture**: Learn about conceptual, external, and internal levels of databases.
* **Indexing**: How to optimize query performance using indexes.
* **SQL Queries**: Write complex SQL queries for data retrieval and manipulation.
* **Keys**: Foreign key, primary key, candidate key, and composite keys.
* **Group By & Having**: Understand how to group data and filter using HAVING clause.
* **Transactions**: Learn how to work with transactions and ensure ACID properties.
* **DML, DDL, DCL**: Master data manipulation (DML), data definition (DDL), and control language (DCL) commands.
* **NoSQL Databases**: Introduction to **MongoDB** or **Cassandra** for handling unstructured data.

**Project**:

* Build a **user management application** with CRUD operations using SQL and integrate with **MongoDB** for unstructured data.

**2. Data Structures & Algorithms**

* **Array, Linked List, String**: Fundamental data structures.
* **Stack, Queue, Hash Table**: Advanced structures for handling data.
* **Tree & Graph**: Deep understanding of binary trees, BST, DFS, BFS, Dijkstra’s algorithm (for route optimization).
* **Trie, Heap**: Use trie for autocomplete features; heap for priority-based tasks like surge pricing.
* **Search Algorithms**: Linear search, binary search.
* **Sorting Algorithms**: Bubble sort, insertion sort, quicksort, mergesort.
* **Recursion**: Solve recursive problems efficiently, keeping time and space complexity in mind.

**Project**:

* Solve **route optimization** problem using **graph algorithms** (Uber application).

**3. Mathematics for Data Science**

* **Linear Algebra**: Matrix operations, vector spaces, eigenvalues/eigenvectors, matrix decomposition.
* **Vector Calculus**: Gradient, divergence, curl, partial derivatives.
* **Probability**: Distributions, Bayes theorem, Markov processes.
* **Statistics**: Descriptive stats, hypothesis testing, p-values, t-tests, confidence intervals, correlation vs. causation.
* **Central Tendency**: Mean, median, mode, variance, standard deviation.

**Project**:

* Perform **demand forecasting** using statistical techniques.

**4. Machine Learning & Optimization**

* **Data Preprocessing**: Handling missing data, feature scaling, encoding categorical variables.
* **Git**: Version control with Git and GitHub.
* **Feature Selection & Engineering**: Techniques to enhance model performance.
* **Regression**:
  + Linear regression, multiple linear regression.
  + Ridge regression, lasso regression, polynomial regression.
  + Decision tree regression, support vector regression (SVR).
* **Evaluation Metrics**: MSE, RMSE, MAE, R-squared.

**Project**:

* **Dynamic pricing model**: Using linear regression and time-series analysis for Uber-style surge pricing.

**5. Classification Algorithms**

* **Logistic Regression**: Binary classification and use of the sigmoid function.
* **Naive Bayes Classifier**: Text classification, spam filtering.
* **K-Nearest Neighbors (KNN)**: Classification for small datasets.
* **Decision Trees & Random Forest**: Building robust classification models.
* **Support Vector Machine (SVM)**: Both linear and non-linear classification.

**Project**:

* Build a **churn prediction model** using decision trees, SVM, and random forest.

**6. Clustering Algorithms**

* **K-Means Clustering**: Unsupervised learning for grouping similar data points.
* **DBSCAN**: Density-based spatial clustering for handling noise.

**Project**:

* Cluster ride demand across **city zones** based on time of day and weather data.

**7. Deep Learning**

* **Neural Networks**: Understand the basics of ANN, backpropagation, and gradient descent.
* **Keras & TensorFlow**: Build and train deep learning models.
* **LSTM Networks**: Use for time series forecasting or sequential data (e.g., predicting ride durations).
* **Convolutional Neural Networks (CNNs)**: Use for computer vision applications (e.g., analyzing traffic camera images).

**Project**:

* Build a **time series model** using **LSTM** to predict **ride demand** at Uber during peak hours.

**8. NLP (Natural Language Processing)**

* **Tokenization & Vectorization**: TF-IDF, Bag of Words, Word2Vec, CountVectorizer.
* **Corpus**: Handling large text datasets.
* **Stemming & Lemmatization**: Preprocessing text for analysis.
* **Named Entity Recognition (NER)**: Extracting important entities like locations, driver names, etc.
* **N-Grams**: Analyzing sequences of words.
* **Advanced NLP**: BERT, GPT, and transformers for complex NLP tasks.

**Project**:

* Perform **sentiment analysis** on Uber’s user reviews and feedback.

**9. Big Data & Distributed Computing**

* **Apache Spark**: Learn Spark for big data processing, including **Spark MLlib** for distributed machine learning.
* **Hadoop**: Understanding of Hadoop architecture for managing large datasets.
* **Apache Kafka**: Learn real-time data streaming for Uber-style applications.

**Project**:

* Build a **real-time ride demand prediction** system using **Apache Kafka** and **Spark** for streaming and processing Uber-scale data.

**10. Data Visualization**

* **Tableau**: Master visualizing data, dashboards, trend lines, and forecasting.
* **PowerBI**: Similar skills with interactive business intelligence dashboards.
* **Custom Visualizations**: Using Plotly, D3.js for customized data visualizations.

**Project**:

* Create a **visual dashboard** showing Uber’s daily ride demand, surge pricing, and revenue trends.

**11. Cloud Computing & Model Deployment**

* **AWS, GCP**: Learn cloud platforms for model deployment, particularly using **SageMaker** or **Google AI**.
* **Docker & Kubernetes**: For containerizing and deploying machine learning models.
* **CI/CD Pipelines**: Automating the deployment of models with Jenkins or CircleCI.

**Project**:

* Deploy a **machine learning model** on **AWS** for real-time dynamic pricing.

**Capstone Project: End-to-End Uber Data Science Application**

**Objective**:

* Build a comprehensive Uber-style data science application involving **demand forecasting**, **route optimization**, and **surge pricing** using machine learning and deep learning techniques.

**Skills Involved**:

* Data preprocessing, SQL, big data handling, machine learning, and deep learning algorithms.
* Real-time data streaming using **Kafka**.
* Data visualization using **Tableau** or **Plotly**.
* Deployment on the cloud using **AWS** or **GCP**.

**Additional Recommendations:**

* **Real-World Problem Solving**: Practice on datasets like Uber’s open-source data for **ride-hailing** or **logistics**.
* **Platform Practice**: Regularly practice on **Kaggle** and **LeetCode**.
* **Networking**: Engage in data science communities on **GitHub**, **Kaggle**, and **LinkedIn**.

With this structured curriculum, you’ll develop a well-rounded skill set and portfolio aligned with the data science role at Uber.