# COURSEPACK

## SCHEME

The scheme is an overview of work-integrated learning opportunities and gets students out into the real world. This will give what a course entails.

<table>
<tr><td>Course Title</td><td colspan="3">Machine Learning Tools and Platforms</td><td colspan="2">Course Type</td><td colspan="3">Integrated</td></tr>
<tr><td>Course Code</td><td colspan="3"></td><td colspan="2">Class</td><td colspan="3">BCA 3rd Sem</td></tr>
<tr><td rowspan="6">Instruction delivery</td><td>Activity</td><td>Credits</td><td>Credit Hours</td><td colspan="4" rowspan="2">Total Number of Classes per Semester</td><td colspan="2">Assessment in Weightage</td></tr>
<tr><td>Lecture</td><td>3</td><td>3</td><td colspan="2">CIE</td><td>SEE</td></tr>
<tr><td>Tutorial</td><td>0</td><td>0</td><td rowspan="3">Theory</td><td rowspan="3">Tutorial</td><td rowspan="3">Practical / Practical</td><td rowspan="3">Self-study</td><td rowspan="3"></td><td rowspan="3"></td></tr>
<tr><td>Practical</td><td>1</td><td>2</td></tr>
<tr><td>Self-study</td><td>0</td><td>0</td></tr>
<tr><td>Total</td><td>4</td><td>5</td><td>45</td><td>0</td><td>15</td><td></td><td>50 %</td><td>50 %</td></tr>
<tr><td>Course Lead</td><td colspan="2">Dr. Ankit Kumar</td><td>Course Coordinator</td><td colspan="5">Mr. Alok Kumar</td></tr>
<tr><td rowspan="2">Names Course Instructors</td><td colspan="3">Theory</td><td colspan="5">Practical</td></tr>
<tr><td colspan="3">Dr. Ashish Tripathi<br>Dr. VDS Baghela<br>Mr. Anshuman<br>Mr. Mujjafar Hussain<br>Dr. Ankit Kumar<br>Dr. Shambhu Jha<br>Mr. Alok Kumar</td><td colspan="5">Dr. Ashish Tripathi<br>Dr. VDS Baghela<br>Mr. Anshuman<br>Mr. Mujjafar Hussain<br>Dr. Ankit Kumar<br>Dr. Shambhu Jha<br>Mr. Alok Kumar</td></tr>
</table>

## COURSE OVERVIEW

Gain a foundational understanding of machine learning concepts and explore the tools and platforms used to develop, train, and deploy ML models. Learn to use popular ML libraries such as Scikit-learn, TensorFlow to implement supervised learning algorithms. Understand key processes such as data preprocessing, model training, evaluation, and hyperparameter tuning. Explore cloud-based machine learning platforms like Google Cloud AI Platform, Amazon SageMaker, and Azure Machine Learning for scalable deployment.

## PREREQUISITE COURSE

| | | |
|---|---|---|
| **PREREQUISITE COURSE REQUIRED** | YES | |
| **If, yes please fill in the Details** | **Prerequisite course code** | **Prerequisite course name** |
| | | **Python Programming** |
| | | **Machine Learning** |

### COURSE OBJECTIVE

1. To introduce students to machine learning tools and environments.
2. To help students implement ML models using Scikit-learn and TensorFlow.
3. To expose students to cloud-based ML platforms and model deployment.

### COURSE OUTCOMES (COs)

After the completion of the course, the student will be able to:

| | |
|---|---|
| | Use basic ML libraries and tools like NumPy, Pandas, and Scikit-learn. |
| | Build simple ML models and evaluate their performance. |
| | Use cloud platforms like Google Colab and Google Cloud for ML tasks. |
| | Create simple web apps for ML using Streamlit or Flask. |

### BLOOM'S LEVEL OF THE COURSE OUTCOMES

**INTEGRATED**

| CO No. | Remember **KL1** | Understand **KL 2** | Apply **KL 3** | Analyse **KL 4** | Evaluate **KL 2** | Create **KL 6** |
|---|---|---|---|---|---|---|
| | | | ✔ | | | |
| | | | ✔ | | ✔ | |
| | | | ✔ | ✔ | | |
| | | | ✔ | | | ✔ |

## PROGRAM OUTCOMES (POs):

| | |
|---|---|
| **PO1** | **Computational Knowledge:** Apply knowledge of computing fundamentals, computing specialization, mathematics, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements. |
| **PO2** | **Problem Analysis:** Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines. |
| **PO3** | **Design /Development of Solutions:** Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations. |
| **PO4** | **Conduct investigations of complex Computing problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |
| **PO5** | **Modern Tool Usage:** Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations. |
| **PO6** | **Professional Ethics:** Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practices. |
| **PO7** | **Life-long Learning:** Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional. |
| **PO8** | **Project management and finance:** Demonstrate knowledge and understanding of the computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. |
| **PO9** | **Communication Efficacy:** Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions. |
| **P10** | **Societal and Environmental Concern:** Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practices. |
| **P11** | **Individual and Team Work:** Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments. |
| **P12** | **Innovation and Entrepreneurship:** Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large. |

## PROGRAM SPECIFIC OUTCOMES(PSOs):

| | |
|---|---|
| **PSO1** | Have the ability to work with contemporary technologies in computing requisite to Industry 4.0 developing and implementing solutions to real life problems. |
| **PSO2** | Demonstrate applications development skills learned through technical training and projects to solve real world problems. |

## COURSE ARTICULATION MATRIX

| COs#/ POs | PO 1 | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO 10 | PO 11 | PO 12 | PSO 1 | PSO 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | 3 | - | - | 1 | - | - | - | - | - | 2 |
| | 2 | - | - | - | 3 | - | 1 | - | - | - | - | - | 3 | 2 |
| | - | 3 | - | - | 3 | - | - | - | - | - | - | - | | |
| | - | - | 3 | - | 3 | 3 | 1 | 3 | 1 | 1 | 3 | 3 | 2 | 3 |

**Note:** 1-Low, 2-Medium, 3-High

## COURSE ASSESSMENT

The course assessment patterns are the assessment tools used both in formative and summative examinations.

| **Type of Course (B)** | **CIE** | | | **Total Marks** | | **Final Marks CIE*0.5+SEE*0.5** |
|---|---|---|---|---|---|---|
| | LAB Work+ Record @ + MoU | MTE | LAB EXAM* | CIE | SEE | |
| INTEGRATED | 25 | 50 | 25 | 100 | 100 | 100 |

@Lab Work-15 marks + MoU WITH WIPRO-10 marks

*Passing Criteria-30% of marks to be secured in the lab Exam conducted by two examiners (one internal and one external)

TEACHING PEDAGOGY: please choose teaching pedagogy according to the COs of subject.

| | INTEGRATED |
|---|---|
| 1 | Interactive Demonstrations |
| 2 | Collaborative Learning |
| 2 | Problem Based Learning |

**SLOW and FAST LEARNER: It is required to complete this process after completion of 25% of syllabus.**

For Integrated Course: Through problem based Learning OR through Lab Experiments

## COURSE CONTENT

THEORY+ PRACTICAL

**Basics of Machine Learning & Tools:** What is Machine Learning? Types of ML (Supervised, Unsupervised), ML workflow and lifecycle, Introduction to Python libraries: NumPy, Pandas, Matplotlib, Working with Jupyter Notebook and Google Colab

**Data Handling and Preprocessing:** Loading datasets (CSV, online), Data cleaning and handling missing values, Feature scaling and encoding, Data splitting (train/test)

**Building ML Models using Scikit-learn:** Introduction to Scikit-learn, Classification: Logistic Regression, K-Nearest Neighbors, Regression: Linear Regression, Clustering: K-Means, Model evaluation: Accuracy, Confusion Matrix

**Introduction to Deep Learning with TensorFlow/Keras:** What is Deep Learning?, Neural network basics, Using Keras to build simple neural networks, Training, validation, and testing, Simple example: MNIST digit classification

**Cloud-based ML Platforms:** Introduction to cloud ML platforms: Google Cloud, AWS, Azure (overview), Using Google Colab with GPU, Intro to Google Cloud AutoML / AWS SageMaker Studio, Creating and training a basic model on cloud

**ML Deployment and Visualization:** Creating a simple web app with Streamlit, Basic API using Flask, Deploying ML models using Heroku or GitHub Pages, Introduction to model monitoring and versioning (basic idea)

## LESSON PLAN FOR INTEGRATED COURSES

**FOR THEORY 15 weeks * 3 Hours = 45 Classes) (1credit = 1Lecture Hour)**
**FOR PRACTICAL 15 weeks * 2Hours = 30 Hours lab sessions (1 credit = 2 lab hours)**

| Lesson No. | Topic | Learning Outcome (LO) | Level (SOLO) | CO Mapping | Assessment | Learning Activity |
|---|---|---|---|---|---|---|
| 1 | Introduction to Machine Learning: Definition, Real-world applications | Describe what machine learning is and identify real-world use cases. | Unistructural | CO1 | Wooclap Quiz | Think-Pair-Share |
| 2 | Types of Machine Learning: Supervised, Unsupervised (with examples) | Differentiate between supervised and unsupervised learning with examples. | Multi-structural | CO1 | Group discussion | Jigsaw |
| 3 | ML Workflow and Lifecycle: Data collection, preprocessing, model building, evaluation, deployment | Explain the stages of the ML lifecycle from data to deployment. | Relational | CO1 | Case study analysis | Think-Pair-Share |
| 4 | Working with Jupyter Notebook | Operate Jupyter Notebook to write and execute Python code. | Unistructural | CO1 | Lab exercise | Interactive demo, Guided exploration |
| 5 | Working with Google Colab: Running code cells, markdown, uploading files | Use Google Colab to run cells, write markdown, and upload files. | Unistructural | CO1 | Lab exercise | Lab exercisePair coding, Demo session |
| 6 | Python for Machine Learning: Basic syntax, variables, loops, functions | Write Python programs using basic syntax, variables, loops, and functions.. | Write Python programs using basic syntax, variables, loops, and functions.Mult | CO1 | Coding test | Think-Pair-Share |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | istructural | | | |
| 7 | Introduction to NumPy: Arrays, basic operations, reshaping | Apply array operations and reshaping using NumPy. | Multi-structural | CO1 | Coding assignment | Collaborative activity |
| 8 | Introduction to Pandas: DataFrames, Series, basic data manipulation | Manipulate data using Series and DataFrames in Pandas.. | Multi-structural | CO1 | Coding assignment | Collaborative activity |
| 9 | Introduction to Matplotlib: Plotting line charts, bar graphs, histograms | Create line charts, bar graphs, and histograms using Matplotlib. | Multi-structural | CO1 | Quiz, Group task | Data plotting pair task |
| 10 | Loading Datasets: Reading CSV files using Pandas, loading from URLs, basic exploration (`head()`, `info()`, `describe()`) | Load and explore datasets using Pandas functions like head(), info(). | Multi-structural | CO2 | Practical exercises | Think-Pair-Share on dataset insights |
| 11 | Data Cleaning: Identifying null/missing values, removing duplicates, replacing or imputing missing data | Identify and clean missing or duplicate data in a dataset. | Multi-structural | CO2 | Lab Task | Fishbowl (approaches to cleaning) |
| 12 | Handling Missing Values: Techniques like mean/median imputation, forward/backward fill, dropping missing entries | Apply techniques to handle missing data appropriately. | Relational | CO2 | Quiz | Problem solving in pairs |
| 13 | Feature Scaling: Normalization (Min-Max), Standardization | Normalize or standardize numerical features in a | Relational | CO2 | Practical quiz | Peer review task |

| | | | | | | |
|---|---|---|---|---|---|---|
| | (Z-score) | dataset. | | | | |
| 14 | Encoding Categorical Variables: One-hot encoding, Label encoding using Pandas | Encode categorical variables using one-hot or label encoding. | Relational | CO2 | Assignment | Guided lab with peer check |
| 15 | Data Splitting: Train-test split, validation sets, random state, stratified sampling | Split data into training and testing sets with control over sampling. | Multi-structural | CO2 | Quiz, Practice task | Think-Pair-Share |
| 16 | Introduction to Scikit-learn: Structure, loading datasets, pipeline overview | Describe Scikit-learn structure and load datasets for modeling. | Multi-structural | CO2 | Quiz, Group Discussion | Jigsaw (dataset APIs) |
| 17 | Classification with Logistic Regression: Model training, prediction, interpretation | Train and interpret a logistic regression classifier using Scikit-learn. | Relational | CO2 | Coding Assessment | Interactive group coding |
| 18 | Classification with K-Nearest Neighbors: Distance metrics, model fitting, prediction | Build a KNN model and analyze results based on distance metrics. | Relational | CO2 | Coding Assessment | Code in pairs, Share outcomes |
| 19 | Model Evaluation I: Accuracy, Confusion Matrix, Precision, Recall | Evaluate models using accuracy, precision, recall, and confusion matrix. | Relational | CO2 | Quiz, Short answer questions | Think-Pair-Share on metrics |
| 20 | Regression with Linear Regression: Fitting models, predicting | Build and evaluate a linear regression model. | Relational | CO2 | Quiz | Think-Pair-Share |
| 21 | Clustering with K-Means: Unsupervised | Perform K-Means clustering and | Relational | CO2 | Practical assessment task | Interactive lab |

| | | | | | | |
|---|---|---|---|---|---|---|
| | learning, visualization | visualize the clusters. | | | | |
| 22 | Model Evaluation II: Cross-validation, overfitting, underfitting | Analyze model performance using cross-validation and interpret overfitting/underfitting. | Relational | CO2 | Quiz, Discussion | Fishbowl debate |
| 23 | Full workflow from data to model evaluation | Integrate data preprocessing, modeling, and evaluation into a complete ML pipeline. | Extended Abstract | CO2 | Practical exercise | Group project |
| 24 | What is Deep Learning? Overview vs traditional ML, applications | Compare deep learning with traditional ML and describe use cases. | Multistructural | CO2,CO3 | Quiz, Case study | Interactive quiz, Discussion pairs |
| 25 | Neural Network Basics: Perceptron, activation functions, layers, forward pass | Explain the structure and working of simple neural networks. | Multi-structural | CO2,CO5 | Quiz, Short answers | Diagram-based peer teaching |
| 26 | Introduction to Keras and TensorFlow: Model structure (Sequential API), compiling models | Construct a basic neural network using Keras Sequential API. | Multi-structural | CO2,CO5 | Lab activity | Hands-on, pair modeling |
| 27 | Training Neural Networks: Epochs, batch size, optimizers, loss functions | Train a neural network using appropriate parameters and observe learning behavior. | Relational | CO2,CO5 | Training log evaluation | Experiment and share results |
| 28 | Validation and Testing: Splitting | Differentiate between | Multistructural | CO2, CO5 | Quiz | Think-Pair-Share |

| | | | | | | |
|---|---|---|---|---|---|---|
| | datasets | training, validation, and testing data splits. | | | | |
| 29 | Example: MNIST Digit Classification – Building and training a CNN | Build and train a CNN to classify handwritten digits from MNIST dataset. | Relational | CO2 | Coding | Group implementation task |
| 30 | Visualizing Results and Model Summary: Accuracy/Loss curves, confusion matrix for DL | Interpret training curves and confusion matrices of neural networks. | Relational | CO2 | Model performance analysist | Compare visualizations in pairs |
| 31 | Overview of Cloud ML Platforms: Google Cloud, AWS, Azure – key services and comparison | Compare ML services offered by Google Cloud, AWS, and Azure. | Multistructural | CO4 | Case study | Jigsaw on cloud platforms |
| 32 | Introduction to Google Colab: Features, advantages, running notebooks, mounting Google Drive | Explain the features and benefits of Google Colab for ML tasks | Multistructural | CO4 | Quiz | Interactive quiz, Peer discussion |
| 33 | Using GPU in Google Colab: Enabling GPU, comparing CPU vs GPU runtime, benchmarking | Enable and benchmark GPU in Colab for ML training. | Relational | CO4 | Practical assignment | Code walkthrough, Peer reviews |
| 34 | Introduction to Google Cloud AutoML: Interface, features, creating datasets and | Use Google Cloud AutoML interface to create and train a model. | Relational | CO3 | Practical assignment | Code walkthrough, Peer reviews |

| | | | | | | |
|---|---|---|---|---|---|---|
| | training models | | | | | |
| 35 | Introduction to AWS SageMaker Studio: Interface, notebooks, model training demo | Demonstrate ML workflow in AWS SageMaker Studio. | Relational | CO4 | Practical assignment | Code walkthrough, Peer reviews |
| 36 | Creating and Training a Basic Model on Google Cloud Platform | Build and deploy an ML model on Google Cloud using AutoML tools. | Relational | CO4 | Quiz | Group activity |
| 37 | Creating and Training a Basic Model on AWS SageMaker Studio | Train and evaluate a model using AWS SageMaker interface. | Relational | CO4 | Peer review, Checklist assessment | Role-playing, Demonstrations |
| 38 | Introduction to ML Deployment: Why deployment matters, overview of deployment options | Explain the purpose of ML deployment and options available. | Multi-structural | CO3 | Practical assignment | Lab exercises, Group work |
| 39 | Streamlit: UI basics, displaying data, integrating ML models | Create a simple UI using Streamlit for ML input/output. | Relational | CO3 | UI demo task | App design pairs |
| 40 | Creating a Simple Web App with Streamlit | Build a basic ML web app using Streamlit. | Relational | CO5 | App build task | Interactive coding pairs |
| 41 | Streamlit – Model Integration and Input Handling: Uploading input, displaying predictions | Integrate ML model with UI and handle user input in Streamlit. | Extended Abstract | CO5 | App build task | Interactive coding pairs |
| 42 | Introduction to Flask: Creating basic API routes, handling requests and responses | Build basic API routes and handle requests using Flask. | Relational | CO5 | Coding task | Peer coding and review |

| | | | | | | |
|---|---|---|---|---|---|---|
| 43 | Building and Testing ML API with Flask: Integrating a trained model, using Postman/cURL | Deploy and test a ML model using Flask and external tools like Postman. | Relational | CO5 | API testing | Group test activity |
| 44 | Deploying ML Apps using Heroku: Setup, deployment steps, troubleshooting | Deploy a Flask app with ML model to Heroku. | Relational | CO5 | Deployment evaluation | Guided deployment session |
| 45 | Introduction to Model Monitoring and Versioning: Concepts, tools overview (MLflow, DVC - intro only) | Describe the basic concepts and tools for model monitoring/versioning. | Multistructural | CO5 | Quiz | Think-pair-share |

**LIST OF PROGRAMS**

| Practical No | Practical |
|---|---|
| 1 | Introduction to Jupyter Notebook / Google Colab: Install Jupyter (if local), open Colab, use Markdown, run basic cells. |
| 2 | Working with NumPy and Pandas: Create arrays, perform slicing, filtering, use DataFrames to import CSV. |
| 3 | Data Cleaning and Preprocessing: Handle missing values, label encoding, one-hot encoding, scaling. |
| 4 | Data Visualization: Plot histograms, bar charts, scatter plots, heatmaps for correlation. |
| 5 | Data Splitting and Model Evaluation Basics: Use train_test_split, calculate accuracy, precision, recall. |
| 6 | Build a Linear Regression Model: Predict house prices or student scores using LinearRegression. |
| 7 | Classification using Logistic Regression: Predict whether a student will pass/fail based on study hours. |
| 8 | K-Nearest Neighbors (KNN) for Classification: Classify iris flower species using KNeighborsClassifier. |
| 9 | Clustering with K-Means: Group customers or animals based on features. |
| 10 | Building a Neural Network using Keras: Use Keras to classify digits using the MNIST dataset. |
| 11 | Google Colab with GPU: Enable GPU and compare training time with |

| | |
|---|---|
| | CPU. |
| 12 | Use of AutoML Tools (Google Cloud / Teachable Machine / H2O.ai): Use Google Teachable Machine or H2O.ai to classify images or data. |
| 13 | Model Deployment using Streamlit:Predict student result or diabetes detection using Streamlit. |
| 14 | Model Deployment with Flask and Heroku:Use Flask to build API; deploy to Heroku (basic template). |
| 15 | Mini Project Presentation: Students present a mini project (data analysis → model → app). |

Text Book (s)


1. Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc", 2022.


Reference Book (s)


1. Raschka, Sebastian, and Vahid Mirjalili. "Python machine learning second edition." *Birmingham, England: Packt Publishing* (2017).
2. McMahon, Andrew P., and Adi Polak. *Machine learning engineering with Python*. Packt Publishing, 2023.
3. Grus, Joel. *Data science from scratch: first principles with python*. O'Reilly Media, 2019.


Webliography:

1. https://learn.microsoft.com/en-us/azure/machine-learning/
2. https://scikit-learn.org/stable/documentation.html
3. https://aws.amazon.com/sagemaker/
4. https://developers.google.com/machine-learning/crash-course

- **PROBLEM-BASED LEARNING**

Exercises in Problem-based Learning (Assignments)

| SN | Problem | KL |
|---|---|---|
| 1 | What is the difference between supervised and unsupervised learning with examples? | KL2 |
| 2 | Design a basic ML pipeline to predict house prices using Linear Regression. | KL4 |
| 3 | Given a dataset with missing values, how will you preprocess it? | KL4 |
| 4 | Compare Logistic Regression and KNN in terms of classification accuracy. | K |
| 5 | Build a model using Scikit-learn to classify Iris flowers. | KL4 |
| 6 | Why is feature scaling important before using KNN? | KL2 |
| 7 | Implement a simple neural network using Keras on MNIST. | KL4 |

| | | |
|---|---|---|
| 8 | What are the key components of the ML lifecycle? | KL1 |
| 9 | Use Pandas to clean a dataset and remove duplicates and null values. | KL4 |
| 10 | Plot a correlation heatmap using Seaborn and analyze relationships. | KL4 |
| 11 | Describe the difference between one-hot encoding and label encoding. | KL2 |
| 12 | Create a scatter plot to visualize the relationship between hours studied and test scores. | KL4 |
| 13 | How would you split a dataset into train/test sets in Scikit-learn? | KL2 |
| 14 | Compare accuracy and confusion matrix as evaluation metrics. | KL5 |
| 15 | Build and deploy a diabetes detection model using Streamlit. | KL6 |
| 16 | What problems arise if we don't normalize features in distance-based algorithms? | KL4 |
| 17 | Train a logistic regression model to predict student pass/fail status. | KL3 |
| 18 | Create a basic Streamlit web app to display prediction results. | KL6 |
| 19 | Describe how GPUs accelerate deep learning model training. | KL2 |
| 20 | What are the key differences between Google Cloud AutoML and AWS SageMaker? | KL2 |
| 21 | Visualize a classification boundary using KNN. What insights can you draw? | KL4 |
| 22 | Use Google Colab to train a model and compare CPU vs GPU training time. | KL3 |
| 23 | Train a clustering model to group customers based on shopping habits. | KL3 |
| 24 | Given a CSV file, write Python code to load and describe its statistics. | KL3 |
| 25 | Use Matplotlib to visualize outliers in a dataset. How would you treat them? | KL4 |
| 26 | Identify common sources of data bias in ML datasets. | KL2 |
| 27 | Describe the difference between underfitting and overfitting. | KL4 |
| 28 | Build an API using Flask to serve predictions from an ML model. | KL6 |
| 29 | What steps are needed to deploy a model on Heroku? | KL2 |
| 30 | Explain how confusion matrix gives more insights than accuracy alone. | KL2 |
| 31 | Explore H2O.ai or Google Teachable Machine to classify hand gestures. | KL3 |
| 32 | Use NumPy to create a matrix and apply slicing and reshaping. | KL3 |
| 33 | What are the pros and cons of cloud-based ML vs local training? | KL5 |
| 34 | Design a model versioning strategy for a production ML app. | KL6 |
| 35 | What is the significance of validation data during training? | KL2 |
| 36 | Train and test a Linear Regression model using only 10% of the dataset. What happens? | KL4 |
| 37 | Simulate a classification problem where classes are imbalanced. How will you handle it? | KL5 |
| 38 | Analyze model performance using precision, recall, and F1-score. | KL4 |

| | | |
|---|---|---|
| 39 | Why do we use activation functions in neural networks? | KL2 |
| 40 | Create a notebook in Google Colab to demonstrate end-to-end ML project. | KL6 |
| 41 | How do you monitor a deployed ML model for data drift? | KL6 |
| 42 | Propose a mini project to predict employee attrition. | KL6 |
| 43 | Integrate a trained ML model into a web interface using Flask and HTML. | KL6 |
| 44 | Modify the learning rate in Keras and analyze its effect on training. | KL4 |
| 45 | Compare K-Means clustering with Hierarchical clustering in terms of output and usage. | KL5 |