-> Scikit-learn :

Main use ML algorithms.

Strength easy ML interface, solid model selection.

Weakness Not for deep learning, smaller datasets.

-> TensorFlow :-

Main use Deep learning.

Strength Scalable GPU, rich DL ecosystem

Weakness steeper learning curve

Summary :-

Panda / Numpy dominate data preprocessing;

Scikit-learn streamlines classical ML;

TensorFlow / PyTorch drive cutting-edge DL projects. Visualization varies, but wrappers and external tools.

6) Data processing and Analytics Workflow

Steps:

1) Raw Data ingestion: Collect data from databases, CSVs, sensors, APIs.

2) Cleaning :- Handle missing values, remove duplicates corrects formats.

3) Transformations :- Scale / normalize features, encode categorical variables.

4) Feature Engineering :- Select / create relevant feature, dimensionality reduction.

5) Model training and selection :- Train / test ML model, use cross-validation for evaluation.

6) Model validation :- Assess Metrics, perform bias check.

7) Visualization and insight generation :- Use plots for exploratory analysis present findings via dashboard.

PART-A : Theoretical Question

1) AI Platform and Tool – Overview and comparison

AI platform streamline workflows for data science, Machine learning and deployment. This section compares Databricks, Google Vertex AI and Azure ML studio:

Databricks :-

Core Compabilities :- Unified workspace for big data, ML, collaborative notebooks; Spark-native.

Strength :- Scable analytics, seamless Spark integratio collaborative features.

Weakness :- Learning curve for advance feature; cost for large cluster.

Ecosystem :- Integrates well with Azure, AWS,

Use case :- ETL, ML model development, data lake

Google vertex AI :-

Core capabilities :- End-to-end M ops, prebuilt models, Mabole infra.

Strengths :- Easy model deployment, powerful AutoML, integrated with Google cloud suite.

Weakness :- less real-time monitoring built-in, pricing for complex ops.

Ecosystem :- Google ecosystem, APIs, GCP service.

Use case :- MLP, computer vision, tabular ML

Azure ML studio :-

Core capabilites :- Drag and drop workflow, integrated model training, deployment, performance traking.

Strength :- User friendly UI, supports Python/R, strong security good for beginning.

Google Vertex AI :-

Core capabilities :- End-to-end M ops, prebuilt models, scalable infra.

Strengths :- Easy model deployment, powerful AutoML, integrated with Google cloud suite.

Weakness :- less real-time monitoring built-in, pricing for complex ops.

Ecosystem :- Google ecosystem, APIs, GCP service.

Use case :- MLP, computer vision, tabular ML

Azure ML Studio :-

Core capabilites :- Drag and drop workflow, integrated model training, deployment, performance tracking.

Strength :- User friendly UI, supports Python/R, strong security good for beginning.

Weakness :- Advanced customization needs code, some feature require premium tier.

Ecosystem :- Integrate with Azure cloud/fabric

Use case :- Predictive analytics, production ML APIs.

Analysis :-

Databricks excels for big data and collaborative analytics ; Vertex AI offers high-level integration and AutoML for production ML pipline ; Azure ML studio is approachable for rapid prototyping and deployment. Choice depends on project scale, required integrations, and team expertise.

2) Prompt Engineering and fine-Tuning :-

Prompt engineering is the art of designing input prompts to improve the accuracy

and relevance of large language Model (LLM) responses. Example: Instead of "summarize the article", a more effective prompt would be: "Summarize the article in bullet points, focusing on climate change impact".

Fine-Tuning:-

Involves further training a base model on domain-specific data to adapt it for specialized tasks.

b) API for AI services — Design and security essentials

To deploy AI model as services, API as created for easy access and integration.

Key design factor include:-

→ Authentication:- Use standard like OAuth2 or JWT to verify clients and protect endpoints.

→ Rate limiting :- Prevent abuse by capping request per time unit.

→ Monitoring :- Track request / response time, usage analytics, error rates.

→ Error Handling :- Provide informative error message and code.

→ Input Validation :- Sanitize, format and validate all incoming data.

Security Best practices :-

→ Use HTTPS to encrypt traffic.

→ Impose strict CORS policies.

→ Isolate model execution and sandbox inference process.

→ Regularly patch and audit software dependencies.

→ Log access and monitor for anomalous usage.

4) Integration AI Models into Applications

Integration workflow:-

→ Model Deployment: Host models as RESTful, gRPC, or direct inference endpoints.

→ Application connectivity: Integrate via API calls or SKDs.

→ Serving Options: Use services (containers, or serverless infrastructure.

→ Monitoring and lifecycle: Track data/model drift, retain/update as needed.

Deployment Methods:

→ Cloud-Based: Highly scalable, easy to maintain.

→ Edge Deployment: Low latency privacy

→ Onprimise:- For data sovereignty, regulatory compliance.

Production Considerations:

Version models, schedule retraining, manage rollback, ensure compatibility with application requirement.

:) Data Science libraries - Architecture and Capabilities.

→ panda :-

Main use is Data Manipulation.

Strength are Fast, Flexible, tabular ops

weakness are Not optimized for very large data

visualization support are Basic plotting via Matplotlib.

→ Numpy :-

Main use Numerical Ops

Strengths Array computation, speed

Weaknesses No direct data table, limited ML