It’s very important to use data, so you can improve the app and user experience.

**Vertex AI** is Google Cloud’s unified platform for building, deploying, and scaling machine learning models. It integrates AutoML, custom training, and MLOps tools into a single environment.

**Machine Learning (ML) and Data Engineering pipeline**: Ingest, Analyse, Transform, Train, Model, Evaluate, Deploy, and Predict. These stages are commonly used in MLOps and data workflows.

**1. Ingest**

* **Purpose**: Bring data from various sources into a centralized system.
* **Sources**: APIs, databases, flat files (CSV, JSON), streaming platforms (Kafka, Kinesis), cloud storage.
* **Tools**: Apache NiFi, AWS Glue, Azure Data Factory, Apache Kafka, Logstash.
* **Best Practices**:
  + Handle schema drift and data quality issues.
  + Use metadata tracking.
  + Automate data ingestion pipelines.

**2. Analyze**

* **Purpose**: Understand the structure, quality, and trends in the data.
* **Activities**: Exploratory Data Analysis (EDA), statistics, data profiling.
* **Tools**: Pandas, Matplotlib, Seaborn, Power BI, Tableau, Jupyter Notebooks.
* **Metrics to Check**:
  + Missing values
  + Outliers
  + Correlations
  + Data distribution

**3. Transform**

* **Purpose**: Clean, normalize, and prepare data for modelling.
* **Activities**:
  + Handle missing values
  + Normalize/standardize data
  + Feature engineering (creating new features)
  + One-hot encoding, label encoding
* **Tools**: Pandas, Spark, dbt, Azure DataBricks, Scikit-learn (preprocessing).
* **Best Practices**:
  + Track transformations.
  + Ensure reproducibility.
  + Separate training and test transformations.

**4. Train**

* **Purpose**: Fit a machine learning algorithm to the training data.
* **Algorithms**: Linear Regression, Decision Trees, Random Forest, XGBoost, Neural Networks.
* **Tools**: Scikit-learn, TensorFlow, PyTorch, XGBoost.
* **Steps**:
  + Choose model type
  + Provide features and labels
  + Fit the model

**5. Model**

* **Purpose**: Represent the learned pattern or function.
* **Artifacts**: Trained weights, model architecture, metadata.
* **Best Practices**:
  + Version your models.
  + Document input/output schemas.
  + Use modular code (pipelines).

**6. Evaluate**

* **Purpose**: Assess model performance and generalization.
* **Metrics**:
  + Classification: Accuracy, Precision, Recall, F1-Score, AUC.
  + Regression: MAE, MSE, RMSE, R².
* **Validation Techniques**: Train/test split, K-fold cross-validation.
* **Tools**: Scikit-learn metrics, MLflow, Azure ML, TensorBoard.

**7. Deploy**

* **Purpose**: Make the trained model available for use in production.
* **Types**:
  + Batch inference
  + Real-time inference (REST API)
* **Tools**: Docker, Kubernetes, Azure ML, AWS SageMaker, TensorFlow Serving, FastAPI.
* **Best Practices**:
  + Automate CI/CD for ML.
  + Monitor performance post-deployment.
  + Implement rollback strategy.

**8. Predict**

* **Purpose**: Use the deployed model to make predictions on new data.
* **Types**:
  + Online (real-time APIs)
  + Offline (batch jobs)
* **Monitoring**:
  + Data drift
  + Model performance decay
  + Latency and throughput
* **Tools**: Model endpoints, scoring scripts, inference pipelines.