

# AI & ML Projects

AI & ML Projects

AI & ML Projects (Combined)

AI & ML Projects (updated)

AI & ML Projects (Updated)

Project 1: World Population Forecast

- Objective: Forecast world population trends using demographic and socio-economic data.
- Methods: Time series modeling, ARIMA, Prophet, LSTM/Transformer sequence models.
- Data sources: UN population estimates, World Bank indicators.

Project 2: Predicting Food Prices in Nigeria

- Objective: Build models to predict food price movements in Nigerian markets.
- Methods: Regression models, tree-based models (XGBoost/LightGBM), temporal models.
- Features: Supply indicators, weather, inflation, transport costs, seasonality.

Project 3: Indian Credit Card / Car Price Predicting

- Objective: Predict credit card default risk and car price/value estimation in Indian markets.
- Methods: Classification for credit risk; regression for car price prediction using features like year, mileage, make/model, region.
- Data sources: Public datasets, scraped market listings, financial records.

Cloud Computing (Added to projects and personal info)

- Overview: Cloud resources and patterns useful for deploying ML workloads and data pipelines.
- Key services to know and practice:
  - Amazon EC2: Launch and manage virtual server instances for model training and inference.
  - Amazon S3: Store datasets, model artifacts, and logs with lifecycle rules and versioning.
  - Amazon VPC: Create isolated networks, subnets, routing tables, security groups for secure deployments.
  - IAM: Manage users, roles, and policies for least-privilege access.
  - Auto Scaling & Load Balancing: Scale model serving and APIs for availability and cost-efficiency.
  - RDS / DynamoDB: Managed databases for metadata and application state.
  - AWS Lambda & Step Functions: Serverless orchestration for lightweight tasks and ETL.
- Suggested hands-on tasks:
  1. Create an S3 bucket, upload sample datasets, enable versioning.

2. Launch an EC2 instance, install Python + ML stack, run a small training job.
3. Set up a VPC with public and private subnets, attach a security group permitting only necessary ports.
4. Create IAM roles for EC2 and for a CI/CD pipeline with least privilege.
5. Deploy a simple model behind an Application Load Balancer with an Auto Scaling Group.

- Notes:

- Follow best practices for credentials: use IAM roles, avoid embedding secrets, rotate keys.
- Consider cost management: use spot instances for training, lifecycle policies for S3.

## Cloud\_Computing\_Info

### Cloud Computing — Practical Guide for Personal Info

This document provides hands-on tasks and brief explanations to add cloud computing skills to your profile.

#### Core Concepts

- EC2: Virtual machines. Use for training small models, experimenting, and running containers.
- S3: Object storage. Store raw data, preprocessed datasets, model checkpoints, and logs.
- VPC: Virtual Private Cloud to isolate resources, create private subnets for databases, public subnets for load balancers.
- IAM: Identity and Access Management for users, groups, roles, and policies.
- Autoscaling & ELB: Keep services available and scale according to load.

#### Getting started (practical steps)

1. Create an S3 bucket; upload a sample CSV dataset; enable versioning and set a lifecycle rule to archive older data.
2. Launch an EC2 instance (Ubuntu 22.04), SSH in, install Python, pip, and common ML libraries (numpy, pandas, scikit-learn, torch/tensorflow).
3. Configure an IAM role for the EC2 instance with S3 read/write permissions (avoid storing AWS keys on the instance).
4. Design a VPC: create subnets, route tables, an internet gateway for public subnet; put databases in private subnet.
5. Deploy a small Flask/FastAPI model server on EC2 and put it behind an Application Load Balancer. Configure an Auto Scaling Group.
6. (Optional) Use AWS Lambda + Step Functions for scheduled ETL jobs that move data from S3 to a database.

#### Security & Cost

- Use Security Groups to restrict inbound traffic.
- Use IAM roles with least privilege.
- Monitor costs; use AWS Cost Explorer and set budgets/alerts.

- Use spot instances for large training jobs to reduce costs.