1. Data Analyst

- **Technical Skills:**
  - **Excel (Advanced):** Pivot tables, formulas (VLOOKUP, INDEX–MATCH), macros.
  - **SQL:** Joins (INNER, OUTER), GROUP BY, window functions (ROW_NUMBER, RANK), subqueries.
  - **Python or R (Optional but Preferred):**
    - **Python:** pandas (dataframes, groupby), NumPy (arrays), matplotlib/seaborn (visualization).
    - **R:** dplyr, tidyr (tidy data), ggplot2 (visualizations).
  - **Visualization Platforms:** Tableau or Power BI—creating interactive dashboards, designing storyboards for reports.
  - **Statistics (Basic):** Mean, median, mode, standard deviation, correlation, p-values, t-tests.
- **Soft Skills:**
  - Business communication (translating data insights into clear recommendations).
  - Report writing (clarity, conciseness).
  - Domain knowledge (e.g., finance, marketing, healthcare) to interpret data contextually.

2. Business Intelligence Developer

- **Technical Skills:**
  - **SQL (Advanced):** Performance tuning, indexing, stored procedures, data partitioning.
  - **ETL Tools:** Informatica, Talend, Microsoft SSIS—designing data pipelines to move data from transactional databases to data warehouses.
  - **BI Tools:**
    - **Tableau:** Data sources, calculated fields, level-of-detail expressions, parameters.
    - **Power BI:** DAX (Data Analysis Expressions), Power Query (M language), dashboard design.

    - **Data Warehousing Concepts:** Dimensional modeling (star/snowflake schemas), ETL vs ELT, OLAP cubes.
    - **Basic Scripting:** Python or shell scripting to automate ETL tasks (optional but helpful).
- **Soft Skills:**
    - Stakeholder liaison (gathering reporting requirements).
    - Visual design principles (choosing appropriate chart types, color theory for dashboards).
    - Analytical thinking (identifying key performance indicators and metrics).

3. Data Scientist

- **Technical Skills:**
    - **Programming:** Python (primary) or R—writing clean, modular code.
        - Python libraries: pandas, NumPy, scikit-learn (for classical ML), matplotlib, seaborn, Plotly (visualization).
        - R libraries: caret, dplyr, ggplot2.
    - **Machine Learning:**
        - **Supervised Learning:**
            - Regression (linear, polynomial, ridge, lasso).
            - Classification (decision trees, random forests, gradient boosting like XGBoost, logistic regression, SVM).
        - **Unsupervised Learning:**
            - Clustering (K-means, hierarchical).
            - Dimensionality reduction (PCA, t-SNE).
        - **Model Evaluation:** Cross-validation, confusion matrix, AUC-ROC, precision/recall, F1 score.
    - **Deep Learning (Introductory):**
        - Familiarity with neural nets—TensorFlow/Keras or PyTorch for image/text tasks. Not mandatory but beneficial as a distinguishing skill.
    - **Statistics & Math:**
        - Descriptive statistics, probability distributions (Gaussian, Poisson, etc.), hypothesis testing, Bayesian basics.
    - **Data Wrangling:**

        - Handling missing values, outliers, categorical encoding, feature scaling/normalization.
    - **SQL/NoSQL:** Extracting data from relational or document databases (MongoDB) for analysis.
    - **Basic Deployment:**
        - Packaging models (pickle/ONNX), building a REST API (Flask, FastAPI) to serve model predictions.
    - **Version Control:** Git for code collaboration.
- **Soft Skills:**
    - Storytelling with data (creating narratives around findings).
    - Collaboration (working with domain experts to understand business context).
    - Experimentation mindset (A/B testing, iterative model improvement).

4. Machine Learning Engineer

- **Technical Skills:**
    - **Strong Programming (Python):** Writing reusable, production-grade code (PEP8, modular design).
    - **Deep Learning Frameworks:**
        - TensorFlow (Keras API, TensorFlow Hub, TFX).
        - PyTorch (nn.Module, DataLoader, training loops).
    - **Model Serving & Deployment:**
        - TensorFlow Serving, TorchServe, ONNX runtime.
        - Build RESTful endpoints (Flask, FastAPI) that load a serialized model (SavedModel, .pt) and return predictions.
        - Containerization: Docker (writing Dockerfiles, managing images).
        - Kubernetes fundamentals: Deploying pods/services, creating Helm charts for model rollout.
    - **MLOps Tools:**
        - CI/CD pipelines (GitHub Actions, Jenkins) for automated training and deployment.
        - Experiment tracking: MLflow, Weights & Biases, TensorBoard.
    - **Cloud Platform Experience:**
        - AWS SageMaker (training jobs, model hosting).
        - GCP AI Platform (training, hyperparameter tuning).

        - Azure ML (workspace, experiments, pipelines).
    - **Data Engineering Basics:**
        - Working with large datasets: Spark (PySpark), Dask.
        - Data storage: S3, GCS, HDFS.
    - **Optimization & Scaling:**
        - Profiling code (cProfile), debugging GPU performance (NVIDIA Nsight).
        - Techniques like mixed-precision training, model quantization for lower-latency inference.
- **Soft Skills:**
    - Problem-solving (adapting models to resource constraints, latency/throughput goals).
    - Cross-functional collaboration (syncing with data engineers, DevOps, and frontend/backend teams).
    - Documentation (maintaining model cards, architecture diagrams, runbooks).

5. Data Engineer

- **Technical Skills:**
    - **SQL Mastery:** Complex joins, window functions, performance tuning (indices, partition pruning).
    - **Big Data Ecosystem:**
        - Apache Hadoop (HDFS, MapReduce paradigm).
        - Apache Spark (RDD vs. DataFrame APIs, PySpark).
        - Kafka (real-time streaming, producers/consumers, partitioning).
    - **Data Warehouse Technologies:**
        - Amazon Redshift, Google BigQuery, Azure Synapse, Snowflake—defining schemas, partitioning, optimizing queries.
    - **ETL/ELT Tools & Orchestration:**
        - Apache Airflow (DAGs, operators, scheduling).
        - AWS Glue, Azure Data Factory.
        - Python or Scala for custom ETL scripts.
    - **Cloud Infrastructure:**
        - AWS (S3 for storage, EMR for Spark clusters), GCP (Cloud Storage, Dataproc), Azure (Blob Storage, Databricks).

    - **Version Control & CI/CD:** Git, GitLab CI or Jenkins for data pipeline workflows.
    - **Data Modeling:** Star schema, snowflake schema, normalized vs. denormalized designs.
- **Soft Skills:**
    - Communication (understanding data requirements from analytics teams).
    - Attention to detail (ensuring data quality, handling edge cases).
    - Troubleshooting (debugging broken pipelines, data inconsistencies).

6. MLOps Engineer

- **Technical Skills:**
    - **Containerization & Orchestration:**
        - Docker: building images, multi-stage builds for small-sized containers.
        - Kubernetes: pods, services, deployments, autoscaling, ConfigMaps.
    - **CI/CD for ML:**
        - GitHub/GitLab Actions or Jenkins: writing pipelines that automatically train and deploy models when new data arrives or code is updated.
        - Automated testing: unit tests for data preprocessing functions, model evaluation tests.
    - **Pipeline Orchestration:**
        - Kubeflow Pipelines, MLflow Projects/Models or TFX (TensorFlow Extended) pipelines.
        - Scheduling and dependencies (ensuring training jobs, preprocessing, and testing happen in the correct order).
    - **Monitoring & Logging:**
        - Prometheus/Grafana dashboards for system health and model performance metrics (accuracy drift, input data distribution changes).
        - ELK Stack (Elasticsearch, Logstash, Kibana) or Splunk for log aggregation.
    - **Infrastructure as Code (IaC):**
        - Terraform: defining cloud resources (EKS clusters, IAM roles) as code.
        - CloudFormation (AWS) or ARM templates (Azure).

    - **Cloud Services (Any One or Multiple):** AWS (SageMaker, EKS, Lambda), GCP (AI Platform, GKE), Azure (ML Pipelines, AKS).
    - **Security & Compliance:** Container scanning (Clair, Trivy), ensuring data encryption at rest/in transit, role-based access control.
- **Soft Skills:**
    - Collaboration (bridge between data scientists and DevOps).
    - Systems thinking (understanding how data flows from ingestion to production inference).
    - Documentation (write runbooks, alert thresholds, rollback procedures).

7. Deep Learning Engineer / Research Engineer

- **Technical Skills:**
    - **Neural Network Architectures:**
        - CNN architectures: AlexNet, VGG, ResNet, EfficientNet for vision tasks.
        - RNNs/LSTMs/GRUs for sequences (speech, text, time series).
        - Transformers (attention mechanism, encoder-decoder, self-attention)—BERT, GPT variants, Vision Transformers (ViT).
    - **DL Framework Mastery:**
        - PyTorch: building custom nn.Module classes, DataLoader for batching, training loops, mixed-precision training (torch.cuda.amp).
        - TensorFlow/Keras: tf.data pipelines, subclassing Model/Layer classes, callbacks, TPU support.
    - **Advanced Topics:**
        - Transfer learning (fine-tuning pretrained models).
        - Data augmentation techniques (CutMix, MixUp, random erasing).
        - Generative models (VAEs, GANs—DCGAN, StyleGAN, CycleGAN) if in GenAI territory.
    - **Mathematical Foundations:**
        - Linear algebra (vectors, matrices, eigenvalues), calculus (chain rule, gradient descent), probability (bayesian inference, KL divergence).
    - **GPU/Distributed Training:**
        - Using multiple GPUs: DataParallel (PyTorch), MirroredStrategy (TensorFlow).
        - Distributed frameworks: Horovod, DeepSpeed, PyTorch Lightning.

    - **Experiment Tracking & Hyperparameter Tuning:**
        - Tools like Weights & Biases, MLflow, or TensorBoard for recording metrics, visualizing loss curves, and comparing runs.
        - Automated hyperparameter search: Optuna, Ray Tune, Hyperopt.
- **Soft Skills:**
    - Research mindset (reading and implementing new papers).
    - Analytical skills (diagnose vanishing gradients, overfitting, underfitting).
    - Collaboration (often part of larger R&D teams, pair with other researchers/engineers).

8. NLP Engineer

- **Technical Skills:**
    - **Text Preprocessing:**
        - Tokenization (WordPiece, SentencePiece), stemming/lemmatization, removing stop words, handling misspellings.
        - Working with unstructured text: cleaning HTML, parsing PDFs, OCR text extraction.
    - **Embeddings & Language Models:**
        - Word embeddings: Word2Vec, GloVe, FastText (training vs. using pretrained).
        - Contextual embeddings: BERT, RoBERTa, GPT, XLNet (fine-tuning for downstream tasks).
        - Sequence labeling (NER), dependency parsing, part-of-speech tagging (spaCy, Stanza).
    - **Advanced Architectures:**
        - Transformer APIs (Hugging Face Transformers: pipelines, Trainer class, tokenizers).
        - Implementing custom transformer blocks (e.g., modifying attention heads, positional encodings).
    - **Model Serving for NLP:**
        - Packaging NLP pipelines into production services (Flask/ FastAPI + Uvicorn/Gunicorn), optimizing inference speed (ONNX, TensorRT).
    - **Evaluation Metrics:**
        - BLEU, ROUGE, METEOR for text generation; F1 score, precision/recall for classification; perplexity for language models.

    - **Speech/NLP Integration:** (Optional, for speech-based products)
        - ASR (Automatic Speech Recognition) frameworks (DeepSpeech, wav2vec), TTS (Text-to-Speech) frameworks.
- **Soft Skills:**
    - Linguistic intuition (understanding context, polysemy, sarcasm).
    - Critical thinking (assessing model biases in language, cultural nuances).
    - Collaboration with domain experts (legal, medical) to tailor NLP solutions.

9. Computer Vision Engineer

- **Technical Skills:**
    - **Image Preprocessing & Augmentation:**
        - OpenCV (image resizing, cropping, thresholding), Pillow (basic PIL operations).
        - Advanced augmentations (Albumentations): random flips, color jitter, random crops, geometric transforms.
    - **Vision Architectures:**
        - CNNs: building blocks (convolutional layers, pooling, batch normalization).
        - Detection models: YOLOv3/v4/v5, Faster R-CNN, SSD—customizing anchor boxes, training on COCO or VOC datasets.
        - Segmentation: U-Net, DeepLab, Mask R-CNN; understanding pixel-level annotation, IoU metrics.
    - **Frameworks & Tools:**
        - PyTorch/TensorFlow for model building.
        - Detectron2 (Facebook's library for object detection), MMDetection (OpenMMLab).
    - **Deployment:**
        - Converting models to TorchScript, TensorRT for edge devices (Jetson Nano, NVIDIA Xavier).
        - Building real-time inference pipelines (OpenCV + C++ or Python streaming).
    - **Evaluation Metrics:**

        - mAP (mean Average Precision) for object detection, Dice coefficient for segmentation, PSNR/SSIM for image quality tasks.
- **Soft Skills:**
    - Attention to detail (labeling bounding boxes precisely).
    - Creative problem-solving (devising augmentation strategies when data is limited).
    - Collaboration with hardware engineers (integrating models into cameras, drones, robotics).

10. Prompt Engineer

- **Technical Skills:**
    - **Deep Familiarity with LLMs & APIs:**
        - OpenAI's API (usage of completions, chat completions endpoints).
        - Azure OpenAI, Anthropic Claude API—knowing version differences (GPT-3.5 vs GPT-4 Turbo vs GPT-4o, Claude 2).
    - **Prompt Crafting Techniques:**
        - Chain-of-Thought prompting, few-shot prompting, zero-shot prompting.
        - "Prompt injection" defense (ensuring safety and robustness).
    - **Scripting Automation:**
        - Python scripts (requests or official SDKs) to automatically send batches of prompts, parse JSON responses, and post-process outputs.
        - Using LangChain or LlamaIndex to build retrieval-augmented generation (RAG) pipelines that combine vector search with LLM prompts.
    - **Basic NLP Understanding:**
        - Knowing tokenization (Byte-Pair Encoding, SentencePiece) to estimate token counts and reduce costs.
    - **Domain-Specific Fine-Tuning/Adapters:**
        - If a project requires fine-tuning on domain text (legal, medical), understanding how to prepare data and run fine-tuning jobs (e.g., OpenAI fine-tuning vs. LoRA on Hugging Face).
- **Soft Skills:**

- Strong written communication (clear, unambiguous prompts).
- Analytical mindset to evaluate model outputs (detect hallucinations, measure relevance).
- Domain knowledge to craft prompts that suit specific industries (e.g., finance, healthcare).

11. Generative AI Researcher / Architect

- **Technical Skills:**
  - **Mathematical Foundations (Advanced):**
    - Deep understanding of probability (Bayesian inference), information theory (KL divergence, cross-entropy), optimization (SGD variants, Adam, RMSprop).
  - **Generative Architectures:**
    - GANs: understanding generator/discriminator training dynamics, stabilizing techniques (Wasserstein GAN, gradient penalty).
    - VAEs: implementing encoder/decoder, KL loss.
    - Diffusion Models: math behind forward diffusion (Markov chain), reverse denoising process, training from scratch.
    - Large-scale Transformers: custom attention mechanisms, memory optimization, sparsity.
  - **Distributed & Scalable Training:**
    - Multi-GPU (DataParallel, DistributedDataParallel in PyTorch).
    - Custom sharding strategies (tensor parallelism, pipeline parallelism) for extremely large models.
    - Frameworks: DeepSpeed, FairScale, Megatron-LM.
  - **Performance Profiling & Optimization:**
    - Using NVIDIA Nsight, PyTorch Profiler, or TensorBoard Profiler to locate bottlenecks.
    - Quantization-aware training, pruning, knowledge distillation to compress models.
  - **Research Workflow:**

        - Versioning experiments (Git + DVC), logging metrics, reproducibility (Docker or Conda environments).
        - Familiarity with GPU clusters or TPUs (e.g., Google Cloud TPU v3/v4).
- **Soft Skills:**
    - Intellectual curiosity (keeping up with ArXiv, participating in ML/AI conferences—NeurIPS, ICML, CVPR).
    - Collaborative research (writing or co-authoring papers, peer reviewing).
    - Mentoring junior researchers, guiding research direction.

12. AI Product Manager

- **Technical Skills (Moderate Coding/Technical Literacy):**
    - **Product Frameworks:** Agile/Scrum (user stories, sprints, backlog grooming).
    - **Basic Data Literacy:** Understanding A/B testing design, interpreting key metrics (conversion rate, uplift), backlog prioritization based on ML feasibility.
    - **High-Level ML Pipeline Understanding:** Data needs (volume, variety), model life cycle (train → validate → deploy → monitor).
    - **Stakeholder Tools:** Jira or Asana for project tracking, Confluence or Notion for documentation.
- **Business/Domain Skills:**
    - Market research (Kano model for feature prioritization), competitive analysis for similar AI products, ROI forecasting.
    - Pricing strategies, go-to-market planning for AI-based features.
- **Soft Skills:**
    - Leadership (coordinating cross-functional teams: engineering, design, marketing).
    - Communication (translating technical trade-offs to business stakeholders).
    - Negotiation (allocating resources, scope-management).