High-level details of AI projects

1. Define a high-value business problem (1 week)
2. Data infrastructure and data engineering analysis (3 days)
3. Mapping the business problem with the AI domain (1 day)
4. Define the solution development approach (2 days)
5. Roadmap/proposal (2 days)
6. Model development based on the selected approach (2 to 16 weeks)
7. Pilot, optional (2-3 months)
8. Go-live (1 month)
9. Monitoring, maintenance, and continuous improvement (ongoing)

1 Define a high-value business problem *(1 week)*

- Initial high-level requirements understanding session
- On-site deep-dive sessions and shadow observations
  - Business process
  - Pain points
  - Current state of data structure, storage et al
  - Current state of technologies and relevant services
- Define high-level project goals and objectives including broad milestones
- Also, validate that the business goal will indeed be achieved if the technology does work

2 Data infrastructure and data engineering analysis *(3 days)*

- Data types
  - Text
  - Image
  - Audio
  - Video et al
- Quality
  - Completeness
  - Validity
  - Accuracy
  - Consistency
  - Integrity
  - Timelines
- Data storage*
  - SQL databases → SQLite, PostgreSQL, MySQL
  - NoSQL databases → Redis, Cassandra, MongoDB
  - Cloud databases → Amazon RDS, Amazon DynamoDB, GCP Cloud SQL, Azure SQL Database
  - Cloud storages → Amazon S3, Azure storage account, GCP bucket
  - Data warehouses → Snowflake, Google BigQuery, Amazon Redshift

* *Data storage selection will depend upon data structure (for examples, relationa/non-relational database, graph databases, documents, key-value stores et al), flexibility goals (do you want to have an option to scale up/down, do you need to accommodate changes in usage or load), and how you want to manage it (do you want to have your team running it or prefer it to be a managed service).*

- Data Pipeline
  - Types

ETL (Extract, Transform, and Load)	ELT (Extract, Load, and Transform)
Data is extracted and transformed in a separate server before loading into the warehouse	Data extraction and transformation happen within the data warehouse directly

It is critical to set up a well designed data pipeline to ensure all your data from various sources would come together in real-time and undergo necessary processing; especially, it would have proper alerts and corrective arrangements for anomalies (example, missing/partial data flow).

  - Steps
    - Step-1: **Data ingestion**
      - Types
        - Real-time data ingestion
        - Batch-based data ingestion
      - Parameters
        - Data velocity
        - Size
        - Frequency
        - Format
    - Step-2: **Data transformation**
      - Steps
        - Data discovery
        - Data mapping
        - Code generation
        - Execution of the code
        - Review
      - Examples of data transformation
        - Aggregation
        - Generalization
        - Integration
        - Manipulation
        - Normalization etc.
    - Step-3: **Data storage**
  - Tools
    - Apache Airflow
    - Apache Spark
    - Apache Hadoop
    - Talend
    - Fivetran
- Data analysis and reporting
- Visualization
- Data governance and ethics
- Security
- What to do if data is not available
  - Check if synthetic data could work
  - Take data collection, cleaning, and preparation as a parallel effort

2A

- Data Lake is a central repository for **raw** and unstructured data.
- Data Warehouse is a central repository of **preprocessed** data for analytics and business intelligence.
- Data Mart is a data warehouse that serves the needs of a **specific business unit**, like a company's finance, marketing, or sales department.

3 Mapping the business problem with the AI domain *(1 day)*

- Generative AI

Computer Vision	Natural language processing (NLP)	Speech recognition
▪ Generating photo-realistic images ▪ Image restoration ▪ Image quality enhancement ▪ Creating 3D models ▪ Creating arts ▪ Generating videos	▪ Generate human-like responses to text-based prompts ▪ Summarize long documents	▪ Voice generation ▪ Creating music

- Discriminative AI

Computer vision	Natural language processing (NLP)	Speech recognition
▪ Image - Image classification - Object detection - Semantic segmentation - Optical character recognition - Hand-written text recognition ▪ Video - Video understanding - Action classification - Video object segmentation	▪ Language modeling ▪ Question answering ▪ Machine translation ▪ Semantic Analysis	▪ Keyword spotting (KWS) ▪ Automatic speech recognition (ASR)

4A How to evaluate the approaches

- Mapped domain
- State-of-the-Art models

4 Define the solution development approach *(1 week)*

- Check if existing AI models serve the purpose
  - Select top candidate pre-trained models for testing
  - Run inference with the real-world data
  - Check accuracy
  - Select the highest performing model, if accuracy meets the requirements
- Fine-tuning existing models with new data
  - Data preparation
  - Select top candidate pre-trained models for training
  - Retrain with custom data using transfer learning
    - Transfer learning is an ML method that uses a pre-trained model as the basis for training a new one.
      - **Inductive** transfer learning is used when labeled data is the same for the target and source domain but the tasks the model works on are different.
      - **Transductive** transfer learning approach is used in scenarios where the domains of the source and target tasks are similar but not exactly the same.
      - **Unsupervised** works similarly to inductive transfer learning. The difference is that the algorithms focus on unsupervised tasks for both source and target tasks.
  - Run inference with the real-world data
  - Check accuracy
  - Select the highest performing model
- Build custom modules on top of existing models
  - Select top candidate pre-trained models
  - Run inference with the real-world data
  - Select the highest performing model
  - Check accuracy
  - Build custom modules according to the client's requirements
- Build from scratch
  - Building model architecture
    - Build a new model architecture/network from the ground up.
    - Modify an existing architecture.
    - Use an existing AI model architecture
  - Data preparation
  - Model training
  - Testing
  - Hyperparameter tuning
  - Loop the last 3 steps until the desired accuracy
  - Select the highest performing version

4Ai Sources for SOTA models

- Google Scholar
  - Research papers
  - Journals
- Conferences
  - Papers
  - Reports
  - Journals

4Aii Trying out the SOTA models

- Two ways
  - Building the model architecture from reading the paper
  - Using the GitHub repository (preferred option, if available)
- Tips
  - Filter the papers by year – check the papers from last 1-year
  - Check H-Index and H5-Index
  - Check license
  - Check if open-sourced
  - Check if GitHub repo is available for
    - Architecture
    - Codes

4Aiii Important conferences

- Computer vision
  - IEEE/CVF Conference on Computer Vision and Pattern Recognition
  - IEEE/CVF International Conference on Computer Vision
  - European Conference on Computer Vision
  - IEEE Transactions on Pattern Analysis and Machine Intelligence
  - IEEE Transactions on Image Processing
  - Pattern Recognition
  - IEEE/CVF Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)
  - Medical Image Analysis
  - IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)
  - International Journal of Computer Vision
- Deep Learning
  - Computer Vision and Pattern Recognition
  - Neural Information Processing Systems
  - International Conference on Computer Vision
  - European Conference on Computer Vision
  - International Conference on Machine Learning
  - AAAI Conference on Artificial Intelligence
  - International Conference on Learning Representations
  - Meeting of the Association for Computational Linguistics
  - Empirical Methods in Natural Language Processing
  - International Joint Conference on Artificial Intelligence
- Speech recognition
  - ICNLSP
  - TSD
  - SPECOM
- NLP
  - ACL: Association for Computational Linguistics
  - EMNLP: Empirical Methods in Natural Language Processing
  - NAACL: North American Chapter of the Association for Computational Linguistics
  - EACL: European Chapter of the Association for Computational Linguistics
  - COLING: International Conference on Computational Linguistics
  - CoNLL: Conference on Natural Language Learning
  - SIGIR: Special Interest Group on Information Retrieval
  - AAAI: Association for the Advancement of Artificial Intelligence
  - ICML: International Conference on Machine Learning
  - ICDM: International Conference on Data Mining
- Similarly, there are reputable conferences on HCI, autonomous vehicles and other fields

5 Roadmap/proposal *(2 days)*

6 Model development and deployment based on the selected approach *(2-4 months)*

6A AI model development details

1. Data collection

- Gathering relevant and high-quality data that is **representative of the problem** at hand.
- This data can be obtained through various sources, such as public datasets, user interactions, or specific data collection campaigns.
- Collected data is generally saved in a previously agreed-upon format and stored in a private and secured database from where we access the data for further processing and annotation.
- Data collection tools
  - **Web Scraping Tools**: BeautifulSoup, Selenium
  - **APIs**: Twitter API, Google Maps API, OpenWeatherMap API
  - **Data Collection Platforms**: Amazon Mechanical Turk, CrowdFlower (now known as Figure Eight)
  - **Data Logging Tools**: Google analytics, Flurry analytics, Mixpanel, Firebase analytics
  - **Surveys and Questionnaires**: Google Forms, SurveyMonkey
  - **Mobile Data Collection Apps**: ODK Collect, KoBoToolbox

2. Data annotation

- Labeling or tagging data to provide **meaningful context** and make it understandable to machine learning algorithms. It involves annotating data with relevant attributes or categories, such as object boundaries, sentiment, or named entities.
- Annotation can be done manually by human annotators or through automated techniques. Data annotation process generally involves multiple levels of supervision to ensure proper annotation. Prior to data annotation, some **preprocessing** is performed to curate the data according to the needs of the machine learning models training requirement
- **Data Labeling Tools**: Labelbox, Amazon SageMaker Ground Truth, Prodigy, VGG Image Annotator (VIA), CVAT, RectLabel, LabelImg, Doccano

3. Data preparation

- Transforms raw data into a suitable format for model training.
- **Data cleaning** eliminates errors, outliers, and inconsistencies, ensuring data quality.
- **Normalization** scales data to a standard range, preventing features with larger values from dominating the model.
- **Feature selection** focuses on choosing relevant features that contribute most to the model's predictive power.
- **Handling missing values** involves strategies like imputation or removal.
- **Data Preparation Tools**: Pandas, NumPy, Scikit-learn, Apache spark, KNIME, RapidMiner, TensorFlow Data Validation(TFDV)

4. Model training

- Steps
  - Choosing proper **hyperparameters**, i.e., epochs, batch size, and learning rate.
  - Choosing an appropriate **optimizer**, i.e., RMSprop, Adam, SGD, etc.
  - Choosing a proper **activation function**, i.e., Softmax, Sigmoid, etc.
  - Choosing the right **loss function**, i.e., Cross-entropy, CTC.
  - Using GPU-powered VMs for training.
- Tools
  - Databricks
  - Amazon Sagemaker
  - Azure ML Studio
  - Google AI Platform
  - For experiment tracking, comparing, monitoring, and registering models

Weights & Biases | Comet ML | neptune.ai | Valohai

5. Testing

- If the desired accuracy is achieved on **training** and **validation** dataset, then the model should now be tested on the **test dataset**.
- Apart from accuracy, there are many metrics that should be checked based on the type of problem solved, i.e., **precision**, **recall**, **wer**, **cer**, **mAP**, etc. Before rolling out the models to production there are some testing methods that should be performed including -

A/B testing | Load testing | Stress testing

6. Deployment

- After the model is tested, it's time for deploying it. There are a few things to consider before deployment.

Real-time/batch inference | CPU/GPU nodes | Throughput requirements | Availability of the application

- **Containerize** the application with Docker. Often, it's a good idea to break the application down into **smaller microservices**. For example, a real-time text recognizer application can have 3 microservices. One to detect text regions in the given image, another to recognize texts in the regions, and the other to accept an API to accept images via requests. The advantage of using this approach is that the microservices can be scaled independently.
- **Deploy the docker** containers with Kubernetes. Kubernetes provides various features like pod autoscaling, node autoscaling, container-wise resource allocation, metrics server, etc. The metric server can be used to monitor various metrics of the cluster, i.e., node memory usage, CPU usage, pod failures, and many others. Cloud services provide built-in monitoring tools like AWS Cloudwatch, Azure monitor, etc which can be used to monitor the cluster. Also, open-source tools like Prometheus, Grafana, etc can be used to monitor many other metrics in real time.
- **CI/CD pipelines** can be used to continuously build, test, and deploy changes to the production environment from git commit. Some of the popular tools are:

GitHub Actions | Jenkins | Argo CD

7 Pilot, optional *(2-3 months)*

8 Go-live *(1 month)*

9 Monitoring, maintenance, and continuous improvement *(ongoing)*

How much does an AI project cost? And how long does it take?

Cost	Timeline	Phase	Output
	3 weeks	**Feasibility**	• AI model (if existing models could fit the purpose) • Approach, methodology, cost, and timeline
	2-4 months	**POC**	AI model
	2-3 months	**Pilot** (optional)	AI model with live results
	1 month	**Go Live**	AI model in production

\+ Data preparation cost might be needed — we'll get to know during feasibility

\+ We recommend taking the intelligence as an output or integrating the API with your enterprise systems. However, if you want to take web or phone applications, that cost would be added — we'll get to know during feasibility