Artificial Intelligence

1. Defining the AI problem

1.1 Identify the problem we are trying to solve using AI (e.g., user segmentation, improving customer service)

- Identify the need that will be addressed
- Find out what information comes in and what output is expected
- Determine whether AI is called for
- Consider upsides and downsides of AI in the situation
- Define measurable success
- Benchmark against domain or organization-specific risks to which the project may be susceptible

1.2 Classify the problem (e.g., regression, unsupervised learning)

- Examine available data (labeled or unlabeled?) and the problem
- Determine problem type (e.g., classifier, regression, unsupervised, reinforcement)

1.3 Identify the areas of expertise needed to solve the problem

- Identify business expertise required
- Identify need for domain (subject-matter) expertise on the problem
- Identify AI expertise needed
- Identify implementation expertise needed

1.4 Build a security plan

- Consider internal access levels or permissions
- Consider infrastructure security
- Assess the risk of using a certain model or potential attack surfaces (e.g., adversarial attacks on real-time learning model)

1.5 Ensure that AI is used appropriately

- Identify potential ways that the AI can mispredict or harm specific user groups
- Set guidelines for data gathering and use
- Set guidelines for algorithm selection from user perspective
- Consider how the subject of the data can interpret the results
- Consider out-of-context use of AI results

1.6 Choose transparency and validation activities

- Communicate intended purpose of data collection
- Decide who should see the results
- Review legal requirements specific to the industry with the problem being solved

2. Managing data to solve the AI problem

2.1 Choose the way to collect data

- Determine type/characteristics of data needed
- Decide if there is an existing data set or if you need to generate your own

INFORMATION
TECHNOLOGY
SPECIALIST

- When generating your own dataset, decide whether collection can be automated or requires user input

2.2 Assess data quality

- Determine if dataset meets needs of task
- Look for missing or corrupt data elements

2.3 Ensure that data are representative

- Examine collection techniques for potential sources of bias
- Make sure the amount of data is enough to build an unbiased model

2.4 Identify resource requirements (e.g., computing, time complexity)

- Assess whether problem is solvable with available computing resources
- Consider the budget of the project and resources that are available

2.5 Convert data into suitable formats (e.g., numerical, image, time series)

- Convert data to binary (e.g., images become pixels)
- Convert computer data into features suitable for AI (e.g., sentences become tokens)

2.6 Select features for the AI model

- Determine which features of data to include
- Build initial feature vectors for test/train dataset
- Consult with subject-matter experts to confirm feature selection

2.7 Engage in feature engineering

- Review features and determine what standard transformations are needed
- Create processed datasets

2.8 Identify training and test data sets

- Separate available data into training and test sets
- Ensure test set is representative

2.9 Document data decisions

- List assumptions, predicates, and constraints upon which design choices have been reasoned
- Make this information available to regulators and end users who demand deep transparency

3. Building an AI model that solves the problem

3.1 Consider applicability of specific algorithms

- Evaluate AI algorithm families
- Decide which algorithms are suitable, e.g., neural network, classification (like decision tree, k means)

3.2 Train a model using the selected algorithm

- Train model for an algorithm with best-guess starting parameters.
- Tune the model by changing parameters
- Gather performance metrics for the model
- Iterate as needed

INFORMATION
TECHNOLOGY
SPECIALIST

3.3 Select specific model after experimentation, avoiding overengineering

- Consider cost, speed, and other factors in evaluating models
- Determine whether selected model meets explainability requirements

3.4 Tell data stories

- Where feasible, create visualizations of the results
- Look for trends
- Verify that the visualization is useful for making a decision

3.5 Evaluate model performance (e.g., accuracy, precision)

- Check for overfitting, underfitting
- Generate metrics or KPIs
- Introduce new test data to cross-validate robustness, testing how model handles unforeseen data

3.6 Look for potential sources of bias in the algorithm

- Verify that inputs resemble training data
- Confirm that training data do not contain irrelevant correlations we do not want classifier to rely on
- Check for imbalances in data
- Guard against creating self-fulfilling prophecies based upon historical biases
- Check the explainability of the algorithm (e.g., feature importance in decision trees)

3.7 Evaluate model sensitivity

- Test for sensitivity of model
- Test for specificity of model

3.8 Confirm adherence to regulatory requirements, if any

- Evaluate outputs according to thresholds defined in requirements
- Document results

3.9 Obtain stakeholder approval

- Collect results and benchmark risks
- Hold sessions to evaluate solution

4. Deploying model in an application

4.1 Train customers on how to use product and what to expect from it

- Inform users of model limitations
- Inform users of intended model usage
- Share documentation
- Manage customer expectations

4.2 Plan to address potential challenges of models in production

- Understand the types of challenges you are likely to encounter
- Understand the indicators of challenges
- Understand how each type of challenge could be mitigated

INFORMATION TECHNOLOGY SPECIALIST

4.3 Design a production pipeline, including application integration

- Create a pipeline (training, prediction) that can meet the product needs (may be different from the experiment)
- Find the solution that works with the existing data stores and connects to the application
- Build the connection between the AI and the application
- Build mechanism to gather user feedback
- Test accuracy of AI through application
- Test robustness of AI
- Test speed of AI
- Test application to fit size of use case (e.g., in AI for mobile applications)

4.4 Support the AI solution

- Document the functions within the AI solution to allow for maintenance (updates, fixing bugs, handling edge cases)
- Train a support team
- Implement a feedback mechanism
- Implement drift detector
- Implement ways to gather new data

5. Monitoring live production

5.1 Engage in oversight

- Log application and model performance to facilitate security, debug, accountability, and audit
- Use robust monitoring systems
- Act upon alerts
- Observe the system over time in a variety of contexts to check for drift or degraded modes of operation
- Detect any way system fails to support new information

5.2 Assess business impact (key performance indicators)

- Track impact metrics to determine whether solution has solved the problem
- Compare previous metrics with new metrics when changes are made
- Act on unexpected metrics by finding problem and fixing it

5.3 Measure impacts on individuals and communities

- Analyze impact on specific subgroups
- Identify and mitigate issues
- Identify opportunities for optimization

5.4 Handle feedback from users

- Measure user satisfaction
- Assess whether users are confused (e.g., do they understand what the AI is supposed to do for them?)
- Incorporate feedback into future versions

5.5 Consider improvement or decommission on a regular basis

- Combine impact observations (e.g., business, community, technology trends) to assess AI value
- Decide whether to retrain AI, continue to use AI as is, or to decommission AI