

AI-102 Master Cheat Sheet

Plan and manage an Azure AI solution (15–20%)

Select the appropriate Azure AI service

1. Computer Vision:

- Azure Cognitive Services - Computer Vision: This service analyzes images and video content, recognizing objects, scenes, faces, and emotions, ideal for image classification, content moderation, and visual search.

2. Natural Language Processing (NLP):

- Azure Cognitive Services - Text Analytics: This service extracts key information from text, identifies sentiment and language, and performs other NLP tasks. Perfect for sentiment analysis, topic modeling, and building chatbots.
- Azure Cognitive Services - Language Understanding: This service goes deeper, providing advanced NLP capabilities like entity recognition, relation extraction, and question answering. Suitable for information extraction and generating insights from text.

3. Decision Support:

- Azure Machine Learning Studio: This low-code/no-code platform lets you build and deploy machine learning models for various tasks, including forecasting, anomaly detection, and classification.
- Azure Databricks: This powerful Apache Spark-based platform enables large-scale data analysis and model training, ideal for complex decision support systems.

4. Speech:

- Azure Cognitive Services - Speech Services: This service transcribes speech to text, translates languages, and generates realistic speech, useful for building voice assistants, speech analytics, and interactive applications.

5. Generative AI:

- Azure OpenAI: This service provides access to powerful pre-trained models like GPT-3 for generating text, translating languages, and writing different creative text formats. Suitable for building innovative applications requiring text generation.

6. Document Intelligence:

- Azure Cognitive Services - Form Recognizer: This service extracts information from structured forms and documents, automating data entry and processing. Great for invoice processing, medical record analysis, and claims processing.
- Azure Cognitive Services - Text Analytics: This service can also analyze documents, extracting key information and sentiment, useful for document understanding and summarization.

7. Knowledge Mining:

- **Azure Databricks:** This platform provides the scalability and analytical power needed for large-scale knowledge mining tasks, uncovering hidden patterns and relationships within datasets.
- **Azure Cognitive Services - Text Analytics:** For smaller datasets, Text Analytics can identify key entities and relations within text, supporting initial knowledge mining efforts.

Plan, create and deploy an Azure AI service

1. Plan for a solution that meets Responsible AI principles:

- **Fairness:** Identify potential biases in your data and chosen algorithms. Use techniques like counterfactual analysis and fairness metrics to evaluate and mitigate bias.
- **Transparency:** Explain how your AI model works and why it makes certain decisions. Provide insights into data sources, algorithms, and limitations.
- **Accountability:** Define clear ownership and responsibility for the development, deployment, and monitoring of your AI solution. Establish processes for addressing unintended consequences or bias.
- **Privacy:** Securely collect, store, and use user data. Implement data anonymization, encryption, and access control mechanisms.
- **Reliability and Security:** Ensure your AI model is robust and resistant to manipulation or misuse. Regularly monitor performance and implement safeguards against security vulnerabilities.

2. Create an Azure AI resource:

- Choose the appropriate service from Azure AI offerings like Text Analytics, Cognitive Services, Machine Learning, or Data Factory.
- Set up the resource by configuring required parameters, connecting to storage accounts, and defining training data (if applicable).
- Follow best practices for resource tagging, access control, and cost optimization.

3. Determine a default endpoint for a service:

- Each Azure AI service provides endpoints for accessing its functionalities.
- Identify the desired endpoint based on your needs, such as production vs. testing environments.
- Configure the default endpoint using tools like Azure Portal, CLI, or SDKs.

4. Integrate Azure AI services into a CI/CD pipeline:

- Integrate your chosen AI service into your existing CI/CD pipeline for automated training, deployment, and testing.

- Utilize tools like Azure DevOps, Logic Apps, or custom scripts to build and deploy ML models.
- Automate model training on new data, evaluation for performance, and deployment to production upon successful validation.

5. Plan and implement a container deployment:

- Containerize your AI model and dependencies using Docker containers or Azure Container Instances (ACI).
- Define a deployment strategy using Azure Kubernetes Service (AKS) or other orchestrators for scaling and high availability.
- Manage container deployments through CI/CD pipelines or Azure App Service for seamless updates and rollbacks.

Additional Resources:

- Microsoft AI: <https://azure.microsoft.com/en-us/solutions/ai>
- Azure AI Services: <https://azure.microsoft.com/en-us/solutions/ai>
- Responsible AI framework: <https://learn.microsoft.com/en-us/azure/machine-learning/concept-responsible-ai?view=azureml-api-2>
- Azure CI/CD: <https://learn.microsoft.com/en-us/azure/devops/?view=azure-devops>

Manage, monitor and secure an Azure AI service

1. Configure diagnostic logging:

- Purpose: To gather detailed information about the operations, performance, and health of your Azure AI resources.
- Steps:
 - Enable diagnostic logging for specific services or resources.
 - Choose the logging category (e.g., activity logs, Logs, performance counters).
 - Specify storage destinations (Azure Storage account, Log Analytics workspace).
- Benefits:
 - Troubleshooting issues.
 - Monitoring resource usage.
 - Auditing activity.

2. Monitor an Azure AI resource:

- Tools: Azure portal, Azure Monitor, resource-specific dashboards.
- Metrics: Track performance, health, and usage (e.g., request latency, error rates, resource utilization).

- Alerts: Set up notifications for critical events or thresholds.

3. Manage costs for Azure AI services:

- Cost Analysis tool: Track costs by service, resource, or tag.
- Budgets: Set spending limits and receive alerts.
- Optimization: Choose appropriate pricing tiers, scale resources, monitor usage patterns.

4. Manage account keys:

- Purpose: Authentication and authorization for accessing Azure AI services.
- Management: Create, regenerate, view, and delete keys.
- Security: Store keys securely, avoid sharing in plain text.

5. Protect account keys using Azure Key Vault:

- Benefits: Centralized, secure storage for keys and secrets.
- Integration: Azure AI services can access keys directly from Key Vault.
- Access control: Restrict access to authorized users and applications.

6. Manage authentication for an Azure AI service resource:

- Options:
 - Azure Active Directory (AD) for fine-grained access control.
 - API keys or tokens for simpler authentication.
- Authorization: Define permissions for different actions and data access.

7. Manage private communications:

- Purpose: Secure communication between resources within a virtual network or using private endpoints.
- Benefits: Enhanced security and privacy for sensitive data.
- Configuration: Set up virtual networks, create private endpoints.

Implement decision support solutions (10–15%)

Create decision support solutions for data monitoring and content delivery

1. Implement a data monitoring solution with Azure AI Metrics Advisor:

Purpose: Continuously monitor your time series data for anomalies and insights.

How it works:

You define thresholds and expected patterns for your data metric.

Metrics Advisor then analyzes incoming data in real-time using AI, detecting deviations from expected patterns.

Alerts are triggered when anomalies are identified, along with explanations of their potential causes.

Benefits:

Proactive anomaly detection prevents performance issues before they impact users.

Root cause analysis helps diagnose and fix problems faster.

Improved data understanding enhances decision-making.

2. Implement a text moderation solution with Azure AI Content Safety:

Purpose: Automatically scan and analyze textual content for potentially harmful or inappropriate language.

How it works:

You configure Content Safety with various filters based on your needs (e.g., profanity, hate speech, sexually suggestive content).

Content Safety then scans text input (posts, comments, chat messages) against these filters.

Content flagged as potentially harmful is either blocked or moderated depending on your chosen configuration.

Benefits:

Ensures a safe and positive online environment for your users.

Reduces risk of reputational damage from unwanted content.

Improves content quality and user experience.

3. Implement an image moderation solution with Azure AI Content Safety:

Purpose: Analyze images for inappropriate visual content, such as violence, nudity, or gore.

How it works:

You configure Content Safety with image-specific filters for prohibited content.

Images are then processed by AI algorithms that identify specific visual patterns.

Flagged images are either blocked or moderated depending on your configuration.

Benefits:

Creates a safe and appropriate visual experience for your users.

Protects your platform from offensive or harmful imagery.

Ensures compliance with content regulations.

Implement computer vision solutions (15–20%)

Analyze images

Including Image Analysis Features in an Image Processing Request

Key steps:

1. Select appropriate features:
 - Azure Computer Vision API: Offers various features for image analysis:
 - Object detection
 - Optical character recognition (OCR)
 - Image tagging
 - Face detection
 - Landmark detection
 - Adult/racy content detection
 - Custom Vision: Build custom models for specialized tasks.
2. Specify features in request:
 - Use API or SDK to construct request, indicating features to apply.
 - Provide image URL or binary data.
 - Customize settings (e.g., confidence thresholds).

Interpreting Image Processing Responses

Key aspects:

1. Structure of response:
 - JSON format with:
 - Overall request status
 - Individual feature results
 - Error messages (if applicable)
2. Understanding feature responses:
 - Object detection: Bounding boxes, labels, confidence scores.
 - OCR: Text extraction, bounding boxes, confidence.
 - Image tagging: Relevant tags, confidence scores.

- Face detection: Bounding boxes, attributes (age, emotion, etc.).

Example (Azure Computer Vision API):

JSON

```
{
  "categories": [
    {
      "name": "people_",
      "score": 0.81542969
    },
    {
      "name": "outdoor_",
      "score": 0.75654675
    }
  ],
  "description": {
    "tags": [
      "person",
      "outdoor",
      "standing",
      "group",
      "smiling",
      "man",
      "woman"
    ],
    "captions": [
      "A group of people standing outdoors smiling."
    ]
  }
}
```

Use code with caution. [Learn more](#)

content_copy

Additional considerations for AI-102:

- Image formats and storage: Azure supports various image formats and storage options (Azure Blob Storage, etc.).
- Responsible AI: Consider privacy, fairness, and bias when using image processing services.

Extracting Text from Images with Azure AI Vision

Azure AI Vision is a powerful tool within the Azure Cognitive Services suite that offers various functionalities for analyzing images. Extracting text from images, also known as optical character recognition (OCR), is one of its key capabilities. Here's how it works:

1. Preparing your image:

- You can upload your image to Azure through various methods, including REST API calls, SDKs, or the Azure portal.
- Supported image formats include common file types like JPEG, PNG, and TIFF.

2. Utilizing Azure AI Vision Text Recognition:

- Once uploaded, you call the Text Recognition API within Azure AI Vision.
- This API analyzes the image and detects any text present.
- Depending on your needs, you can choose between detecting printed text or handwritten text with specific options.

3. Receiving the extracted text:

- The API returns the extracted text as structured data. This data can include:
 - The recognized text itself.
 - Information about the location and bounding boxes of each text element within the image.
 - Confidence scores indicating the certainty of the recognized text.

4. Integrating the extracted text:

- Once you have the text, you can integrate it into your application in various ways, such as:
 - Populating text fields in forms.
 - Indexing the text for search purposes.
 - Analyzing and understanding the content of the extracted text.

Benefits of using Azure AI Vision for text extraction:

- High accuracy: Microsoft claims high accuracy and performance for both printed and handwritten text recognition.
- Flexibility: Supports various languages and character sets, as well as different text styles and layouts.

- Scalability: Can handle large volumes of images with ease.
- Customization: Offers control over text recognition behavior through various API parameters.

Converting Handwritten Text with Azure AI Vision

Extracting text from handwritten documents within images presents an additional challenge. Fortunately, Azure AI Vision provides functionalities specifically designed for this task.

1. Fine-tuning for handwritten text:

- When performing text recognition, you can specify that the image contains handwritten text.
- This activates specialized algorithms within the API, optimizing it for better accuracy with handwritten characters.

2. Training custom models:

- For even better performance with specific types of handwriting or documents, you can train custom models using the Azure Cognitive Services Custom Vision service.
- This allows you to provide the API with training data consisting of similar handwritten documents to improve its recognition accuracy for such scenarios.

3. Utilizing ink recognition features:

- Beyond simply converting the text, Azure AI Vision offers further tools like ink recognition.
- This analyzes the strokes and shapes of the handwriting, allowing you to extract additional information like pen strokes, underlines, and annotations.

Benefits of using Azure AI Vision for handwritten text conversion:

- Improved accuracy: Specialized algorithms and custom models lead to better recognition of handwritten characters.
- Versatility: Handles various handwriting styles and document types.
- Advanced features: Extracting more than just plain text through ink recognition analysis.

Implement custom computer vision models by using Azure AI Vision

1. Choose between image classification and object detection models:

- Image classification: Identifies the overall category of an image (e.g., cat, dog, car).
- Object detection: Locates and identifies individual objects within an image (e.g., multiple cats and dogs in an outdoor scene).

Choosing the right model depends on your specific task. If you just need to know what the entire image is about, use classification. If you need to find and identify specific objects within the image, use object detection.

2. Label images:

Labeling involves creating data for your AI model to learn from. You annotate images with relevant information, such as object names, bounding boxes, or attributes. Accurate and thorough labeling is key for training powerful and accurate models.

3. Train a custom image model, including image classification and object detection:

Azure Cognitive Services provides tools like Custom Vision to train your own image classification and object detection models. You upload your labeled images, configure training parameters, and the service optimizes the model's performance.

4. Evaluate custom vision model metrics:

Metrics like accuracy, precision, and recall assess your model's effectiveness. Analyze these metrics to understand how well your model performs and identify areas for improvement.

5. Publish a custom vision model:

Once trained and evaluated, you can publish your model for production use. This allows other applications and services to integrate and leverage your AI capabilities.

6. Consume a custom vision model:

After publishing, you can "consume" your model in various ways. Integrate it into your applications to analyze images in real-time, trigger actions based on detected objects, or enhance user experiences with computer vision insights.

Analyze videos

1. Azure AI Video Indexer:

- Function: Extracts various insights from videos, including:
 - Scenes and shots: Automatically segments the video into meaningful units based on content changes.
 - Entities: Recognizes and indexes people, places, and organizations appearing in the video.
 - Keywords and concepts: Identifies key topics and themes discussed in the video.
 - Transcripts and captions: Generates automatic transcripts and captions, including speaker identification and sentiment analysis.
 - Custom insights: Allows extending the analysis with your own models and classifiers.
- Benefits:
 - Streamlines video indexing and search within large libraries.
 - Enables content discovery and retrieval based on semantic understanding.
 - Enhances accessibility and searchability for visually impaired users.
 - Provides valuable metadata for further analysis and visualization.

- Use cases: Video search and navigation, content moderation, education and training, market research, media analysis, and more.

2. Azure AI Vision Spatial Analysis:

- Function: Analyzes video recordings using computer vision to detect human presence and movement:
 - People counting: Tracks the number of people entering, exiting, or remaining within a designated area.
 - Heatmaps: Creates visual representations of people's movement patterns within a space.
 - Crowd analysis: Estimates crowd density and identifies potential bottlenecks.
 - Line crossing detection: Tracks individuals entering or leaving specific areas.
- Benefits:
 - Provides real-time insights into crowd behavior and activity.
 - Improves resource allocation and operational efficiency in spaces like retail stores, airports, and public events.
 - Enables security applications by detecting unauthorized entry or loitering.
 - Offers valuable data for marketing and customer behavior analysis.
- Use cases: Retail customer traffic analysis, crowd management in public spaces, security and surveillance systems, queue optimization, and marketing campaign effectiveness evaluation.

Relationship between the two services:

While these services offer distinct functionalities, they can be used together for a more comprehensive analysis.

- Video Indexer can provide the context and semantic understanding of the video content, while Vision Spatial Analysis adds another layer by focusing specifically on people and their movements.
- Combining insights from both services can generate valuable data for various applications, like understanding how specific demographics interact with video content or analyzing crowd behavior during different scenes.

Implement natural language processing solutions (30–35%)

Analyze text by using Azure AI Language

1. Extract Key Phrases:

- This task involves identifying the most important and representative phrases within a piece of text.
- Key phrases capture the main ideas and topics without going into unnecessary detail.
- Techniques used for key phrase extraction include:

- Frequency analysis: Identifying words and phrases that appear frequently.
- Statistical methods: Using natural language processing (NLP) algorithms to assess the linguistic and semantic importance of phrases.
- Keyword extraction: Identifying phrases related to specific topics or domains.

2. Extract Entities:

- Entity extraction focuses on identifying and classifying named entities within text, such as people, places, organizations, dates, and numbers.
- Entities are key data points that can be used for further analysis and tasks like information retrieval and question answering.
- Common entity extraction methods include:
 - Rule-based systems: Employing predefined rules and patterns to recognize entities.
 - Machine learning models: Trained on large datasets of annotated text to automatically identify entities.
 - Gazetteers: Matching text against lists of known entities (e.g., geographical locations, company names).

3. Determine Sentiment of Text:

- Sentiment analysis aims to understand the emotional tone and opinion expressed in a piece of text.
- It categorizes text as positive, negative, neutral, or mixed based on the underlying subjective content.
- Sentiment analysis uses NLP techniques like:
 - Lexicon analysis: Matching words and phrases to dictionaries with assigned sentiment scores.
 - Machine learning models: Trained to recognize sentiment patterns in text.
 - Rule-based systems: Identifying sentiment-related expressions and grammatical cues.

4. Detect the Language Used in Text:

- Language detection involves automatically identifying the language used in a piece of text.
- This is crucial for tasks like machine translation, information retrieval, and targeted advertising.
- Common language detection methods include:
 - Character n-gram analysis: Analyzing the frequency of character sequences to identify language patterns.
 - Word n-gram analysis: Analyzing the frequency of word sequences to match known language vocabulary.

- Machine learning models: Trained on large datasets of text in different languages.

5. Detect Personally Identifiable Information (PII):

- PII detection focuses on identifying and anonymizing sensitive information in text, such as names, addresses, social security numbers, and email addresses.
- This is essential for complying with data privacy regulations and protecting user information.
- PII detection uses techniques like:
 - Regular expressions: Matching textual patterns indicative of specific PII types.
 - Rule-based systems: Employing predefined rules to identify and classify PII data.
 - Machine learning models: Trained to recognize PII patterns in text with high accuracy.

Process speech by using Azure AI Speech

1. Implement Text-to-Speech:

This involves converting written text into spoken audio. Azure offers various Text-to-Speech services, allowing you to:

- Choose different voices and languages.
- Control speaking rate and pitch.
- Add emphasis and emotions.
- Integrate with applications and websites.

2. Implement Speech-to-Text:

This converts spoken audio into written text. Azure Speech Services provide features like:

- Real-time transcription of conversations.
- Punctuation and capitalization for accuracy.
- Support for various audio formats and accents.
- Integration with translation and other AI services.

3. Improve Text-to-Speech with SSML:

Speech Synthesis Markup Language (SSML) allows fine-grained control over the synthesized speech. You can use SSML tags to:

- Specify pronunciation.
- Add pauses and breaks.
- Change speaking style (whispering, shouting).
- Insert emphasis and emotions.

4. Implement Custom Speech Solutions:

For specific needs, you can create custom speech models trained on your domain-specific data. This improves accuracy for specialized vocabulary and terminology. Azure Speech Services provide tools for:

- Data collection and annotation.
- Model training and deployment.
- Continuous improvement through feedback.

5. Implement Intent Recognition:

This identifies the user's intention behind their spoken words. Azure LUIS service helps you:

- Define intents and entities (specific information within the speech).
- Train a LUIS model with your labeled data.
- Integrate LUIS into your applications to understand user goals.

6. Implement Keyword Recognition:

This detects specific keywords or phrases within spoken language. Azure Speech Services allow you to:

- Define a list of keywords relevant to your application.
- Identify when these keywords are spoken in real-time.
- Trigger actions based on keyword recognition.

Translate language

1. Translating Text and Documents with Azure AI Translator:

- Azure AI Translator is a cloud-based service that uses machine learning to translate text and documents into over 70 languages.
- Developers can integrate Translator into their applications using REST APIs or SDKs available in various programming languages.
- Features include:
 - High-quality translations by leveraging neural machine translation models.
 - Custom language settings to adjust translation nuances.
 - Batch translation for large volumes of text and documents.
 - Terminology dictionary support for domain-specific language accuracy.

2. Implementing Custom Translation:

- Azure AI Translator allows you to train your own custom translation models for specific domains or languages.

- This involves uploading relevant training data, including parallel corpora (source and target language pairs), and training the model on the chosen platform (Azure Machine Learning or Custom Translator Toolkit).
- You can then improve the model's performance by providing feedback and retraining with additional data.
- Finally, you can publish your custom model for use in your applications or share it with others.

3. Speech-to-Speech Translation with Azure AI Speech:

- Azure AI Speech provides real-time translation of spoken language between pairs of supported languages.
- You can integrate this capability into your applications using SDKs or speech APIs.
- Features include:
 - Two-way simultaneous translation for natural conversations.
 - Speaker diarization to identify and track individual speakers.
 - Custom voice models for improved accuracy with specific speakers or accents.

4. Speech-to-Text Translation with Azure AI Speech:

- Azure AI Speech can also transcribe spoken language into text and translate it into another language.
- This enables scenarios like speech-to-text captions for audio/video content or real-time translation of spoken meetings.
- Features include:
 - Highly accurate speech recognition in various languages and accents.
 - Punctuation and speaker diarization options for better understanding.
 - Integration with Translator for seamless speech-to-text translation workflow.

5. Translating to Multiple Languages Simultaneously:

- Azure AI Translator allows you to translate text or speech into multiple languages concurrently.
- This can be useful for scenarios where content needs to be accessible to a diverse audience.
- You can specify the target languages you want to translate to, and the service will handle the translation process.

Implement and manage a language understanding model by using Azure AI Language

Intents and Utterances:

- Intents: Represent the user's goals or intentions behind their utterances.

- **Utterances:** The various ways a user might express those intents in natural language.

Creating intents and adding utterances:

1. **Identify key intents:** What actions or information do users want from your application?
2. **Group similar utterances:** Collect diverse examples of how users might express each intent.
3. **Consider variations:** Account for different phrasings, synonyms, misspellings, and slang.
4. **Add utterances to intents:** Build a comprehensive set of examples for each intent.

Entities:

- **Represent key information:** Extract specific values (e.g., names, dates, locations) from user utterances.

Creating entities:

1. **Define entity types:** Determine what types of information are relevant to your application.
2. **Specify entity roles:** Indicate how entities relate to intents (e.g., required, optional).
3. **Provide examples:** Guide the model in recognizing entities correctly.

Training, Evaluating, Deploying, and Testing a Language Understanding Model:

Training:

1. **Provide labeled data:** Use a dataset of intents, utterances, and entities to train the model.
2. **Choose a training algorithm:** Select an appropriate model architecture and training method.
3. **Monitor training progress:** Track metrics like accuracy and loss to assess model performance.

Evaluating:

1. **Measure performance:** Use a separate test set to evaluate the model's accuracy and generalization.
2. **Identify areas for improvement:** Analyze errors to pinpoint areas where the model needs refinement.

Deploying:

1. **Make the model accessible:** Integrate the trained model into your application or service.
2. **Consider hosting options:** Choose a suitable deployment environment (e.g., cloud, on-premises).

Testing:

1. **Validate functionality:** Ensure the deployed model performs as expected in real-world scenarios.
2. **Gather user feedback:** Collect data on user interactions to further refine the model.

Optimizing a Language Understanding Model:

Techniques:

1. Add more training data: Improve accuracy and coverage by expanding the training dataset.
2. Adjust model parameters: Fine-tune hyperparameters to optimize performance.
3. Apply feature engineering: Extract relevant features from text data to enhance model understanding.
4. Experiment with different algorithms: Explore alternative model architectures and training methods.

Consuming a Language Model from a Client Application:

Integration:

1. Use APIs or SDKs: Communicate with the model from your application using provided tools.
2. Process user input: Send user utterances to the model for intent and entity recognition.
3. Interpret model responses: Handle model output to extract intent, entities, and confidence scores.
4. Trigger appropriate actions: Based on the model's understanding, initiate relevant actions in your application.

Backup and Recovery of Language Understanding Models:

Importance:

1. Protect against data loss: Regularly back up models to prevent loss due to errors or failures.
2. Enable model rollback: Facilitate recovery to previous versions if needed.

Methods:

1. Use provided tools: Utilize platform-specific backup and recovery features.
2. Implement custom solutions: Develop scripts or tools for backing up and restoring models.

Create a question answering solution by using Azure AI Language

1. Create a Question Answering Project:

- This first step defines the scope and purpose of your conversational AI. You'll determine the domain of expertise and target audience to tailor the questions and answers.

2. Add Question-and-Answer Pairs Manually:

- This involves feeding your AI examples of questions and their corresponding answers. Manual entry is a good starting point, but remember to diversify the phrasing and complexity of questions to improve understanding.

3. Import Sources:

- Expand your knowledge base by importing existing data sources like documents, datasets, or other conversational systems. This provides rich material for training and improves the AI's ability to answer diverse questions.

4. Train and Test a Knowledge Base:

- Use the prepared data to train the AI's model. This typically involves algorithms like machine learning to analyze the question-answer pairs and identify patterns. Testing ensures the model performs well and can accurately answer questions.

5. Publish a Knowledge Base:

- Once trained, deploy your knowledge base to make it accessible for users. This could involve integrating it into a chat interface, website, or even mobile app, allowing users to interact with your conversational AI.

6. Create a Multi-Turn Conversation:

- Go beyond simple one-question-one-answer interactions. Design your AI to handle extended conversations, understand context, and respond accordingly. This requires considering dialogue flow and managing user intent across multiple turns.

7. Add Alternate Phrasing:

- Train your AI to recognize and answer the same question phrased differently. This increases user satisfaction and ensures understanding even when users don't use the exact wording from your knowledge base.

8. Add Chit-Chat to a Knowledge Base:

- Make your AI more personable and engaging by adding light, informal conversation capabilities. This could involve greetings, small talk, or humor, making the interaction feel more natural and enjoyable.

9. Export a Knowledge Base:

- Allow others to benefit from your work by exporting your knowledge base in a compatible format. This enables sharing with collaborators, integration with other systems, or even open-sourcing for further development.

10. Create a Multi-Language Question Answering Solution:

- Expand your reach by making your AI multilingual. Train your model with data in multiple languages and develop mechanisms for language detection and translation, empowering your AI to answer questions from users globally.

Implement knowledge mining and document intelligence solutions (10–15%)

Implement an Azure Cognitive Search solution

Provisioning a Cognitive Search resource

- Involves creating an Azure Cognitive Search service instance within your Azure subscription.
- This service provides the foundation for building AI-powered search experiences within your applications.

Creating data sources

- Data sources define connections to the content you want to make searchable.
- Supported sources include Azure Blob Storage, Azure SQL Database, Cosmos DB, and more.

Creating an index

- Indexes are structured containers that store and organize searchable content.
- Define searchable fields (like text, numbers, dates) and their data types during index creation.
- Indexes are optimized for fast retrieval of information.

Defining a skillset

- Skillsets apply AI-powered enrichment to content during indexing.
- Use built-in skills for language detection, text analysis, key phrase extraction, and more.
- Create custom skills using JavaScript for specialized processing tasks.

Implementing custom skills and including them in a skillset

- Extend Cognitive Search functionality with custom skills written in JavaScript.
- Integrate machine learning models or custom logic for unique processing needs.

Creating and running an indexer

- Indexers connect data sources to indexes and execute the content processing pipeline.
- Define an indexer to specify the data source, target index, skillset, and scheduling options.
- Run the indexer to ingest and enrich content for search.

Querying an index

- Use a REST API or SDKs to execute search queries against your indexes.
- Formulate queries using search terms, filters, sorting, and other query syntax elements.
- Retrieve relevant results from the indexed content.

Managing Knowledge Store projections

- The Knowledge Store is a separate storage layer for enriched content and metadata.
- Create file, object, or table projections to access and utilize this data independently of search.
- Useful for knowledge mining, content exploration, and other scenarios.

Implement an Azure AI Document Intelligence solution

Provision a Document Intelligence resource:

- This involves setting up your Document Intelligence service within Azure. You'll choose a pricing tier, region, and configure access control.

- You can provision the service through the Azure portal, Azure CLI, or SDKs.

Use prebuilt models to extract data from documents:

- Prebuilt models are ready-made solutions within Document Intelligence trained for specific tasks like extracting key information from invoices, forms, or receipts.
- You can use the "Analyze" API to pass your document to the chosen prebuilt model and receive the extracted data in JSON format.
- This is a quick and easy way to start using Document Intelligence without building your own models.

Implement a custom document intelligence model:

- If you need to extract data specific to your document types or tasks, you can build your own custom model.
- This involves creating labeled training data with relevant fields you want to extract, choosing your model architecture (e.g., Layout V2, Form Recognizer), and training the model.
- Once trained, the model can be used with the "Analyze" API to extract data from your documents.

Train, test, and publish a custom document intelligence model:

- Training involves feeding your labeled data into the chosen model architecture and iteratively improving its accuracy.
- You can evaluate your model's performance with a separate test dataset and fine-tune its parameters.
- Once satisfied, you can publish the model for production use and integrate it into your workflows.

Create a composed document intelligence model:

- Composed models combine multiple prebuilt and/or custom models in a sequence to extract complex data from documents.
- You can define the processing order and configure how each model's output feeds into the next.
- This is useful when your documents contain various types of information that need different extraction techniques.

Implement a document intelligence model as a custom Azure Cognitive Search skill:

- Azure Cognitive Search allows you to enrich your search index with data extracted from various sources, including Document Intelligence.
- You can create a custom skill based on your document intelligence model and associate it with your search index.
- When users search the index, the skill automatically extracts relevant data from the indexed documents and adds it to the search results for comprehensive retrieval.

Implement generative AI solutions (10–15%)

Use Azure OpenAI Service to generate content

1. Provisioning an Azure OpenAI Service resource:

- Purpose: This involves creating a resource within Azure that provides access to OpenAI's powerful generative AI models.
- Steps:
 1. Access the Azure portal.
 2. Search for "Azure OpenAI Service" and create a new resource.
 3. Select a subscription, resource group, and region for deployment.
 4. Provide the necessary information, including a name for the resource.
 5. Review and agree to the terms of use.
 6. Create the resource.

2. Selecting and deploying an Azure OpenAI model:

- Purpose: This involves choosing the appropriate OpenAI model for your specific use case and making it available for use.
- Steps:
 1. Access the Azure OpenAI Service resource you created.
 2. Choose from available models (e.g., text-davinci-003, code-davinci-002).
 3. Consider factors like model capabilities, cost, and latency when selecting.
 4. Deploy the chosen model.

3. Submitting prompts to generate natural language:

- Purpose: This involves using the deployed model to generate text, such as text completion, translation, or writing different creative text formats.
- Methods:
 1. Use the Azure OpenAI Service REST API.
 2. Use Azure OpenAI Service SDKs (available for various programming languages).
 3. Provide a prompt (text input) to guide the model's text generation.

4. Submitting prompts to generate code:

- Purpose: This involves using the model to generate code snippets in response to prompts, potentially aiding in programming tasks.
- Methods:
 1. Similar to natural language generation, use the REST API or SDKs.

2. Provide a prompt that specifies programming language, desired functionality, and constraints.

Additional considerations:

- Cost: Pricing varies based on model size and usage.
- Responsible AI: Use models ethically and responsibly, considering bias and potential misuse.
- Security: Implement appropriate measures to protect sensitive data.

Optimize generative AI

1. Configuring Parameters to Control Generative Behavior:

- Purpose: To tailor the model's outputs to your specific needs and preferences.
- Key Parameters:
 - Temperature: Controls the degree of creativity and randomness in the responses (higher temperature = more creative and surprising, lower temperature = more factual and consistent).
 - Response Length: Specifies the desired length of the generated text.
 - Top P: Filters out less likely, potentially repetitive text sequences.
 - Stop Sequences: Define phrases that signal the model to stop generating text.

2. Applying Prompt Engineering Techniques to Improve Responses:

- Purpose: To guide the model toward generating more relevant, informative, and high-quality responses.
- Effective Techniques:
 - Clearly define the task or question within the prompt.
 - Provide examples or context to improve understanding.
 - Use keywords and phrases that align with the desired output style.
 - Break down complex prompts into smaller, more manageable steps.

3. Using Your Own Data with an Azure OpenAI Model:

- Purpose: To personalize the model's responses and make them more relevant to your specific domain or use case.
- Process:
 1. Gather and format your data appropriately.
 2. Upload it to Azure OpenAI.
 3. Fine-tune the model using your data (explained in topic 4).

4. Fine-Tuning an Azure OpenAI Model:

- Purpose: To further adapt the model to your specific needs and improve its performance on your tasks.
- Process:
 1. Choose a pre-trained model from Azure OpenAI.
 2. Provide a dataset of text examples and corresponding labels (if applicable).
 3. Train the model on this dataset to adjust its parameters and improve its ability to generate relevant responses.

Disclaimer: All data and information provided on this site is for informational purposes only. This site makes no representations as to accuracy, completeness, correctness, suitability, or validity of any information on this site & will not be liable for any errors, omissions, or delays in this information or any losses, injuries, or damages arising from its display or use. All information is provided on an as-is basis.