COURSE GLOSSARY

Python Toolbox

calling the function does not execute its body immediately but returns a generator to iterate over

chunksize: An argument (commonly used with pandas.read_csv) that specifies the number of rows to read per chunk so large files can be processed incrementally

DataFrame: A two-dimensional, tabular data structure in pandas with labeled axes (rows and columns) used to store and manipulate datasets

Dictionary comprehension: Similar to a list comprehension but using curly braces and a key: value output expression to build dictionaries from iterables in a compact form

enumerate: A built-in function that takes an iterable and returns an iterable of index-value pairs, optionally starting indexing at a specified start value

File object (file connection): An object representing an open file that is iterable over its lines and can be read incrementally using iteration, iter(), or next().

Generator expression: A comprehension-like syntax using parentheses instead of brackets that returns a generator object which yields items on demand

Deployment strategies: Approaches for releasing new model versions into production—such as basic (full replace), shadow (parallel testing), and canary (gradual rollout)—that balance risk, cost, and validation needs

git branch (command): A Git command used to list, create, rename, or delete branches depending on the flags and arguments provided

items() method: A dictionary method that returns an iterable view of the dictionary's (key, value) pairs, often used to loop over both keys and values

iter(): A built-in function that takes an iterable and returns its corresponding iterator object so you can manually retrieve elements

Iterable: An object capable of returning its elements one at a time, typically by providing an __iter__ method so it can be used in for loops and other iteration contexts

Iterator protocol: The pair of conventions in Python (the __iter__ and __next__ methods) that an object must implement to be iterable and to function as an iterator

Iterator: An object that produces successive values when asked, implementing a __next__ method (or next in Python 2) and maintaining internal state across calls

Lazy evaluation: A strategy where values are computed only when needed, reducing memory use and potentially delaying expensive computation until required

List comprehension: A concise syntax for creating a list by specifying an output expression and a for-clause (and optional conditionals) inside square brackets, often replacing simple for loops

Modulo operator (%): An arithmetic operator that returns the remainder of division between two numbers, commonly used to test properties like evenness (n % 2 == 0)

next(): A built-in function that retrieves the next value from an iterator and advances its internal state, raising StopIteration when no values remain

pandas.read_csv: A pandas function that reads CSV files into DataFrame objects and can return iterable chunked readers when provided a chunksize for memory-efficient processing

StopIteration: An exception raised by an iterator to signal that there are no more values to produce and iteration should stop

Unpacking (star operator): The use of the * operator to expand an iterable's elements into individual values in function calls or data structures, consuming the iterable as it unpacks

yield: A keyword used inside generator functions to produce a value and pause the function's state so it can be resumed to produce subsequent values

zip: A built-in function that accepts multiple iterables and returns an iterator of tuples, where each tuple contains the i-th elements from each input iterable