COMP204 Final Cheat Sheet

by cp2300 via cheatography.com/103341/cs/21267/

Binary Numbers

$89_{10} = ?_2$

	result	remainder
divide 89 by two	44	1
divide 44 by two	22	0
divide 22 by two	11	0
divide 11 by two	5	1
divide 5 by two	2	1
divide 2 by two	1	0
divide 1 by two	0	1

So: $89_{10} = 1011001_2$.
Check : $89_{10} = (1 \times 2^6)+(0 \times 2^5)+(1 \times 2^4)+(1 \times 2^3)+(0 \times 2^2)+(0 \times 2^1)+(1 \times 2^0)$.

Binary Numbers

8 bits	1 byte
16 bits	word
32 bits	double word
always at least 1 byte	add one to first number in sequence to make negative, add zeros to get full byte

Machine Learning

-program adjusts itself automatically to fit data, end result is a program trained to achieve a given task

Supervised learning-given examples of input and desired outputs, predict outputs on future unseen inputs(classification, regression, time series)

Unsupervised learning-creates a new representation of the input

Reinforcement learning-learning action to maximize payoff

Types of supervised learning tasks

1. classification- predict which predefined set of classes and example belongs to

2. regression - predict a real value

2. probability estimation - estimate probability of an event

Sensitivity = fraction of positive examples predicted to be positive

TP/(TP+FN)

Specificity= proportion of negative examples predicted negative

TN/(FP+TN)

Machine Learning (cont)

False-positive rate(FPR)= negatives predicted to be positive

FP/(FP+TN)

Sets

Set	stores an unordered set of objects, CANT BE INDEXED, no duplicates, only immutable objects contained
myset = set([])	creates a set from a list
len(myset)	length of the set
if x in myset:	evaluates boolean condition
for element in myset:	iterate through set
for set1 = A and set2 = B, A&B = inters-ection	elements in common to A and B
A | B or A.union(B)	union of two sets
A-B	difference of sets, elements in A that are not in B
	all these rules can be applied to multiple sets

Integer Sequences

range(start, end, step)	(start default is zero, step default 1)
range(5)	0 1 2 3 4
range(3,8)	3 4 5 6 7
range(len-(seq))	sequence of index of values in seq
range(2,12,3)	2 5 8 11

Functions

def functi-onname(arguments):	defines a function of given name with given arguments
return	only returns a certain value or string generated by the function, doesn't print, exits at this value
list = [[0 for i in range(-ncols)] for j in range(nro-wns)]	creates a two dimensional list of nrows and ncols filled with zeros

Operations on Lists

lst = []	creates empty list
lst.appen-d(val)	adds item to list at end
lst.exten-d(seq)	add sequence of items at end
lst.insert(idx, val)	inserts item at index
lst.remov-e(val)	remove first item in the list with value val
lst.pop(idx)	remove and return item at index
lst.sort()/lst.r-everse()	sort/reverse list in place
lst.mi-n()/.max()	finds the min/max

matplotlib.pyplot

.plot(x,y, color)	plots the x values to x values in a certain color
.show()	shows the graph
.ylabe-l(name)	names y axis
.xlabe-l(name)	names x axis
.savefig(figname)	saves image under figname

OOP

class	defines attributes(information we want to keep together) and methods(operations want to be able to perform on that data)
class some_class_name:	defines a class of some_class_name
def __int__(self, other attributes):	defines and initializes attributes
object.method()	calls method on an object of that class

Bugs

Sytax errors	violation of a writing rule
Exceptions(runtime)	syntax is fine, some other thing is occasionally flawed:ZeroDivisionError(can't divide by zero), NameError-(can't access the variable), IndexError(When the index you are attempting to access does not exist), TypeError(operation on non-compatible type)
Logical (semantic) errors	code runs, but doesn't do what its supposed to
try:	does something that may cause an exception
except <SomeExceptionType>:	do something to handle the exception
raise[exception object]	raises a certain, defined type of exception

Bugs (cont)

try, except, else	try something, that could cause the exception, if its fine, go to else
finally:	adds statement that happens no matter what
for a, b in zip(list 1, list 2):	iterates over elements of two lists in parallel, yields a tuple with both iterations

File Input/Output

f = open(myfile, 'x')	opens the file, x=r for reading only, x=w for writing only, x=a for appending, x=b for file in binary format, x=wr+ reading and writing and so on
.read(-size)	reads the entire file, returns a single string, size is optional number of characters
.readlines(size)	reads all lines and returns them as a list of strings
.rstrip()	returns a string with the end-of-line character(\n) removed from the end of the string
.readline()	reads single line from file, returns empty string if end of file
.close()	closes file
gzip.open()	interface to read/write compressed files, .gz extension

Operations on Dictionaries

dict = {}	sets up an empty dictionary
dict[key] = value	sets value of said at that key to value
dict[key]	calls the value at that key
del dict[key]	deletes a key from dict
dict.clear()	clears the dict of all keys
dict.update(dict2)	can update/add associations from another
dict.keys()/-dict.items()/dict.values()	looks at either all keys, values, or all items in the dict
dict.pop(key)	removes element associated to that key
dict.popitem()	removes key, value pair and returns it as a tuple
dict.get(key)	returns value at that key
dict.setdefault(key)	either returns value of the key or creates the given key if not previously existent