CSD

Projects

The following list of projects is taken from the industry to equip learners with as much practical working experience as possible. Each learner has to choose 10 assignments and 3 projetcs from the following list of programming assignments/projects

- Write a python program to take two inputs from a user (a number and its base) and convert that number to a decimal number. Base can be 2 (binary) or 8 (octal) or 16 (hexadecimal)
- Write a python program to determine if one string is a rotation of another. For example, "hello" has the following rotated versions: "lohel", "ohell", and "elloh", but "elolh" is not a rotation because the last three letters are out of order. User can input two strings and the program determines if one string is a rotation of another or not.
- Write a python program to ask user to input a date in DD-MM-YYYY format and print calendar for that month.
- Create a simple text file with 10 common words and name it mydictionary.txt and write a python program to print scores of those words if they were scrabble words. The official scrabble score will be addition of each letter score to get the word score. The letter scores are:

  letter_value = {'a':1 , 'b':3, 'c':3, 'd':2, 'e':1, 'f':4. 'g':2, 'h':4, 'i':1, 'j':8, 'k':5, 'l':1, 'm':3, 'n':1. 'o':1, 'p':3, 'q':10, 'r':1, 's':1, 't':1, 'u':1, 'v':8, 'w':4, 'x':8, 'y':4, 'z':10}

Projects

- **Customer ratings of 100,000 restaurants was given as a JSON file at https://gknxt.com/python/data/yelp_directory.json Write a python program print all restaurants that have 4 or better rating in 76050 area (Dallas, Texas)**

- **You are given data about airline fights in the format of a list of lists, where each list represents one flight. In particular, each list has five strings: the identifying flight information (which is two words, the airline and a number), the three letter airport code for the departing city, the three letter airport code for the arrival city, the number of seats on the plane, and the estimated number of minutes in the air.**

  datalist = [['Delta 165', 'RDU', 'ATL', '172', '50'], ['JetBlue 1862', 'RDU', 'DTW', '190', '109'], ['Southwest 175', 'RDU', 'DEN', '220', '235'], ['American 1567', 'RDU', 'DEN', '290', '232'], ['JetBlue 4576', 'DTW', 'DEN', '190', '190'], ['Delta 526', 'ATL', 'RDU', '78', '55'], ['Southwest 562', 'ATL', 'DEN', '290', '200'], ['American 1274', 'RDU', 'DEN', '290', '232'], ['Delta 1452', 'PHX', 'ATL', '350', '209'], ['Southwest 157', 'DTW', 'ATL', '260', '115'], ['American 237', 'RDU', 'DEN', '451', '192'], ['Delta 275', 'RDU', 'ATL', '50', '90'], ['JetBlue 422', 'DTW', 'PHX', '340', '160']]

  Write the function **longFlights** which has three parameters, datalist, that is a nonempty list of lists of five strings in the format given, a string airline representing an airline, and an integer time. This function returns a sorted unique list of departure cities for airline whose flights are longer than time

Projects

- **Consider the following data file of information on buying items at a grocery store. Each line in the file represents one purchase by a customer. The format of each line is in one of two formats depending on whether the item has a price or is sold by weight and thus has a weight and a price per pound. The first item is the customerID:item : P or W. If the letter is P, it is price, W is weight (price per kg)**

45623:apples:W:2.4:5.00

7634:peanut butter:P:4.00

45623:plums:W:1.5:2.50

45623:spinach:W:1.0:3.50

2375:eggs:P:5.20

7634:oats:W:0.5:3.00

45623:oj:P:3.75

7634:bananas:W:3.2:1.50

2375:yogurt:P:3.25

Write the function groceryPurchases that has three parameters, data, custid and amount, where data is the list of lists in the format rgiven, custid is a customer id, and amount is a the amount of money the customer has.

This function determines which items the customer can purchase based on the desired items to purchase in data and the amount of money they have. That is, this function returns a list of tuples representing items the customer wanted and has enough money to purchase.

Projects

- The entire text from all 37 plays written by William Shakespeare was given as a txt file at https://gknxt.com/python/data/shakespeare.txt Using that file, find the longest word and most frequent word ever written by Shakespeare.

- Write code that includes a list comprehension to assign to the variable temp a list of the strings from the list words in which each word from the original list that is of length four or more has the rst and last letter swapped. The resulting list should have the modified words in the same order as the original list.

  words = ['yellow', 'purple', 'green', 'blue', 'red', 'pink']

  For example, if words was the list above, then after executing the list comprehension, then temp would be the list ['welloy', 'eurplp', 'nreeg', 'elub', 'kinp']

- In USA, a small deposit is added to drink containers to encourage people to recycle them. In one city, drink containers holding one liter or less have a $0.10 deposit, and drink containers holding more than one liter have a $0.25 deposit.

  Write a program that reads the number of containers of each size from the user. Your program should continue by computing and displaying the refund that will be received for returning those containers. Format the output so that it includes a dollar sign and always displays exactly two decimal places.

Projects

- At a bank, a new savings account that earns 4 percent interest per year. The interest that a customer earns is paid at the end of the year, and is added to the balance of the savings account. Write a program that begins by reading the amount of money deposited into the account from the user. Then the program should compute and display the amount in the savings account after 1, 2, and 3 years. Display each amount so that it is rounded to 2 decimal places.

- In the United States, fuel efficiency for vehicles is normally expressed in miles-per gallon (MPG). In Canada, fuel efficiency is normally expressed in liters-per-hundred kilometers (L/100 km). Use your research skills to determine how to convert from MPG to L/100 km. Then create a program that reads a value from the user in American units and displays the equivalent fuel efficiency in Canadian units.

- The surface of the Earth is curved, and the distance between degrees of longitude varies with latitude. As a result, finding the distance between two points on the surface of the Earth is more complicated than simply using the Pythagorean theorem.
  Let (t1, g1) and (t2, g2) be the latitude and longitude of two points on the Earth's surface. The distance between these points, following the surface of the Earth, in kilometers is:
  distance = $6371.01 \times \arccos(\sin(t1) \times \sin(t2) + \cos(t1) \times \cos(t2) \times \cos(g1 - g2))$
  The value 6371.01 in the previous equation wasn't selected at random. It is the average radius of the Earth in kilometers.
  Write a python program that allows the user to enter the latitude and longitude of two points on the Earth in degrees. Your program should display the distance between the points, following the surface of the earth, in kilometers.

Projects

- Python includes a library of functions for working with time, including a function called asctime in the time module. It reads the current time from the computer's internal clock and returns it in a human-readable format. Write a program that displays the current time and date. Your program will not require any input from the user.

- Most years have 365 days. However, the time required for the Earth to orbit the Sun is actually slightly more than that. As a result, an extra day, February 29, is included in some years to correct for this difference. Such years are referred to as leap years. The rules for determining whether or not a year is a leap year follow:
  - Any year that is divisible by 400 is a leap year.
  - Of the remaining years, any year that is divisible by 100 is not a leap year.
  - Of the remaining years, any year that is divisible by 4 is a leap year.
  - All other years are not leap years.

  Write a program that reads a year from the user and displays a message indicating whether or not it is a leap year.

- A parity bit is a simple mechanism for detecting errors in data transmitted over an unreliable connection such as a telephone line. The basic idea is that an additional bit is transmitted after each group of 8 bits so that a single bit error in the transmission can be detected.
  Parity bits can be computed for either even parity or odd parity. If even parity is selected then the parity bit that is transmitted is chosen so that the total number of one bits transmitted (8 bits of data plus the parity bit) is even. When odd parity is selected the parity bit is chosen so that the total number of one bits transmitted is odd.
  Write a program that computes the parity bit for groups of 8 bits entered by the user using even parity. Your program should read strings containing 8 bits until the user enters a blank line. After each string is entered by the user your program should display a clear message indicating whether the parity bit should be 0 or 1. Display an appropriate error message if the user enters something other than 8 bits.

Projects

- A string is a palindrome if it is identical forward and backward. For example "anna", "civic", "level" and "hannah" are all examples of palindromic words. Write a program that reads a string from the user and uses a loop to determines whether or not it is a palindrome. Display the result, including a meaningful output message.

- Words like first, second and third are referred to as ordinal numbers. In this exercise, you will write a function that takes an integer as its only parameter and returns a string containing the appropriate English ordinal number as its only result. Your function must handle the integers between 1 and 12 (inclusive). It should return an empty string if a value outside of this range is provided as a parameter. Include a main program that demonstrates your function by displaying each integer from 1 to 12 and its ordinal number. Your main program should only run when your file has not been imported into another program.

- One of the first known examples of encryption was used by Julius Caesar. Caesar needed to provide written instructions to his generals, but he didn't want his enemies to learn his plans if the message slipped into their hands. As result, he developed what later became known as the Caesar Cipher.
  The idea behind this cipher is simple (and as a result, it provides no protection against modern code breaking techniques). Each letter in the original message is shifted by 3 places. As a result, A becomes D, B becomes E, C becomes F, D becomes G, etc. The last three letters in the alphabet are wrapped around to the beginning: X becomes A, Y becomes B and Z becomes C. Non-letter characters are not modified by the cipher.
  Write a program that implements a Caesar cipher. Allow the user to supply the message and the shift amount, and then display the shifted message. Ensure that your program encodes both uppercase and lowercase letters. Your program should also support negative shift values so that it can be used both to encode messages and decode messages.

Projects

- Write a function that generates a random password. The password should have a random length of between 7 and 10 characters. Each character should be randomly selected from positions 33 to 126 in the ASCII table. Your function will not take any parameters. It will return the randomly generated password as its only result. Display the randomly generated password in your file's main program. Your main program should only run when your solution has not been imported into another file.

- An integer, n, is said to be perfect when the sum of all of the proper divisors of n is equal to n. For example, 28 is a perfect number because its proper divisors are 1, 2, 4, 7 and 14, and $1 + 2 + 4 + 7 + 14 = 28$.

  Write a function that determines whether or not a positive integer is perfect. Your function will take one parameter. If that parameter is a perfect number then your function will return true. Otherwise it will return false. In addition, write a main program that uses your function to identify and display all of the perfect numbers between 1 and 10,000.

- When analysing data collected as part of a science experiment it may be desirable to remove the most extreme values before performing other calculations. Write a function that takes a list of values and an non-negative integer, n, as its parameters. The function should create a new copy of the list with the n largest elements and the n smallest elements removed. Then it should return the new copy of the list as the function's only result. The order of the elements in the returned list does not have to match the order of the elements in the original list.
  Write a main program that demonstrates your function. Your function should read a list of numbers from the user and remove the two largest and two smallest values from it. Display the list with the outliers removed, followed by the original list. Your program should generate an appropriate error message if the user enters less than 4 values.

Projects

- In order to win the top prize in a particular lottery, one must match all 6 numbers on his or her ticket to the 6 numbers between 1 and 49 that are drawn by the lottery organizer. Write a program that generates a random selection of 6 numbers for a lottery ticket. Ensure that the 6 numbers selected do not contain any duplicates. Display the numbers in ascending order.

- Tokenizing is the process of converting a string into a list of substrings, known as tokens. In many circumstances, a list of tokens is far easier to work with than the original string because the original string may have irregular spacing. In some cases substantial work is also required to determine where one token ends and the next one begins.

  In a mathematical expression, tokens are items such as operators, numbers and parentheses. Some tokens, such as *, /, ^, (and) are easy to identify because the token is a single character, and the character is never part of another token. The + and - symbols are a little bit more challenging to handle because they might represent the addition or subtraction operator, or they might be part of a number token.

  Write a function that takes a string containing a mathematical expression as its only parameter and breaks it into a list of tokens. Each token should be a parenthesis, an operator, or a number with an optional leading + or - (for simplicity we will only work with integers in this problem). Return the list of tokens as the function's result.

Projects

- Write a function named reverseLookup that finds all of the keys in a dictionary that map to a specific value. The function will take the dictionary and the value to search for as its only parameters. It will return a (possibly empty) list of keys from the dictionary that map to the provided value.

  Include a main program that demonstrates the reverseLookup function as part of your solution to this exercise. Your program should create a dictionary and then show that the reverseLookup function works correctly when it returns multiple keys, a single key, and no keys. Ensure that your main program only runs when the file containing your solution to this exercise has not been imported into another program.

- Pig Latin is a language constructed by transforming English words. While the origins of the language are unknown, it is mentioned in at least two documents from the nineteenth century, suggesting that it has existed for more than 100 years. The following rules are used to translate English into Pig Latin:

  • If the word begins with a consonant (including y), then all letters at the beginning of the word, up to the first vowel (excluding y), are removed and then added to the end of the word, followed by ay. For example, computer becomes omputercay and think becomes inkthay.

  • If the word begins with a vowel (not including y), then way is added to the end of the word. For example, algorithm becomes algorithm way and office becomes office way.

  Write a program that reads a line of text from the user. Then your program should translate the line into Pig Latin and display the result. You may assume that the string entered by the user only contains lowercase letters and spaces.

Projects

- Write a program that determines and displays the number of unique characters in a string entered by the user. For example, Hello, World! has 10 unique characters while zzz has only one unique character.

- While the popularity of cheques as a payment method has diminished in recent years, some companies still issue them to pay employees or vendors. The amount being paid normally appears on a cheque twice, with one occurrence written using digits, and the other occurrence written using English words. Repeating the amount in two different formsmakes it much more difficult for an unscrupulous employee or vendor to modify the amount on the cheque before depositing it.
  In this exercise, your task is to create a function that takes an integer between 0 and 999 as its only parameter, and returns a string containing the English words for that number. For example, if the parameter to the function is 142 then your function should return "one hundred forty two". Use one or more dictionaries to implement your solution rather than large if/elif/else constructs. Include a main program that reads an integer from the user and displays its value in English words.

- The novel "Gadsby" is over 50,000 words in length. While 50,000 words isn't normally remarkable for a novel, it is in this case because none of the words in the book use the letter "e". This is particularly noteworthy when one considers that "e" is the most common letter in English.
  Write a program that reads a list of words from a file and determines what proportion of the words use each letter of the alphabet. Display the result for all 26 letters. Include an additional message identifying the letter that is used in the smallest proportion of the words. Your program should ignore any punctuation marks and it should treat uppercase and lowercase letters as equivalent.

Projects

- The baby names data set consists of over 200 files. Each file contains a list of 100 names, along with the number of times each name was used. There are two files for each year: one containing names used for girls and the other containing names used for boys. The data set includes data for every year from 1900 to 2012.
  Write a program that reads every file in the data set and identifies all of the names that were most popular in at least one year. Your program should output two lists: one containing the most popular names for boys and the other containing the most popular names for girls. Neither of your lists should include any repeated values.

- There is at least one word in the English language that contains each of the vowels a, e, i , o, u and y exactly once and in order. Write a program that searches a file containing a list of words and displays all of the words that meet this constraint. The user will provide the name of the file that will be searched. Display an appropriate error message and exit the program if the user provides an invalid file name or if something else goes wrong while searching for words with six vowels in order.

- A spell checker can be a helpful tool for people who struggle to spell words correctly. In this exercise, you will write a program that reads a file and displays all of thewords in it that are misspelled. Misspelled words will be identified by checking each word in the file against a list of known words. Any words in the user's file that do not appear in the list of known words will be reported as spelling mistakes.
  The user will provide the name of the file to check for spelling mistakes as a command line parameter. Your program should display an appropriate error message if the command line parameter is missing. An error message should also be displayed if your program is unable to open the user's file. The words followed by a comma, period or other punctuation mark are not reported as spelling mistakes. Ignore the capitalization of the words when checking their spelling.

Projects

- Write a function, most_repeating_word, that takes a sequence of strings as input. The function should return the string that contains the greatest number of repeated letters. In other words
  1) For each word, find the letter that appears the most times.
  2) Find the word whose most-repeated letter appears more than any other.

- Create a program that determines whether or not it is possible to construct a particular total using a specific number of coins. For example, it is possible to have a total of $1.00 using four coins if they are all quarters. However, there is no way to have a total of $1.00 using 5 coins. Yet it is possible to have $1.00 using 6 coins by using 3 quarters, 2 dimes and a nickel. Similarly, a total of $1.25 can be formed using 5 coins or 8 coins, but a total of $1.25 can not be formed using 4, 6 or 7 coins.
  Your program should read both the dollar amount and the number of coins from the user. It should display a clear message indicating whether or not the entered dollar amount can be formed using the number of coins indicated. Assume the existence of quarters, dimes, nickels and pennies when completing this problem. Your solution must use recursion. It can not contain any loops.

- The edit distance between two strings is ameasure of their similarity—the smaller the edit distance, the more similar the strings are with regard to the minimum number of insert, delete and substitute operations needed to transform one string into the other. Consider the strings kitten and sitting. The first string can be transformed into the second string with the following operations: Substitute the k with an s, substitute the e with an i, and insert a g at the end of the string. This is the smallest number of operations that can be performed to transform kitten into sitting.
  As a result, the edit distance is 3. Write a recursive function that computes the edit distance between two strings.

Projects

- You are given a spreadsheet that contains a list of N athletes and their details (such as age, height, weight and so on). Write a python program to sort the data based on the K^{th} attribute and print the final resulting table. An example given below:

Rank	Age	Height (in cm)		Rank	Age	Height (in cm)
1	32	190		5	24	176
2	35	175	sort based on k=1	4	26	195
3	41	188	→	1	32	190
4	26	195	i.e (age)	2	35	175
5	24	176		3	41	188

If two attributes are the same for different rows, for example, if two athletes are of the same age, print the row that appeared first in the input. The input format is:

The first line contains N and M separated by a space.

The next N lines each contain M elements.

The last line contains K.

- A newly opened multinational brand has decided to base their company logo on the three most common characters in the company name. They are now trying out various combinations of company names and logos based on this condition. Given a string S, which is the company name in lowercase letters, write a python program to find the top three most common characters in the string.

Print the three most common characters along with their occurrence count.
Sort in descending order of occurrence count.
If the occurrence count is the same, sort the characters in alphabetical order.

Projects

- You are given n words. Some words may repeat. Write a python program to output its number of occurrences for each word in the input. The output order should correspond with the input order of appearance of the word. Each input line ends with a "\n" character.

  The sum of the lengths of all the words do not exceed 10^6

  All the words are composed of lowercase English letters only.

  Input Format: The first line contains the integer. The next lines each contain a word.

  Output Format: On the first line, output the number of distinct words from the input.

  On the second line, output the number of occurrences for each distinct word according to their appearance in the input.

- Write a python program that takes two groups of words (m words in Group A and n words in Group B) as input (words may repeat) and print the indices of each occurrence of words of group B in group A. The first line of input contains integers, m and n separated by a space.

  The next m lines contains the words belonging to A group .

  The next n lines contains the words belonging to B group .

- The Cypress Tree goes through 2 cycles of growth every year. The first growth cycle occurs during the spring, when it doubles in height. The second growth cycle occurs during the summer, when its height increases by 1 meter. Now, a new Cypress Tree sapling is planted at the onset of spring. Its height is 1 meter. Write a python program to find the height of the tree given N growth cycles?

Projects

- There are two kangaroos on an x-axis ready to jump in the positive direction (i.e, toward positive infinity). The first kangaroo starts at location x1 and moves at a rate of v1 meters per jump. The second kangaroo starts at location x2 and moves at a rate of v2 meters per jump. Write a python program that takes starting locations and movement rates for each kangaroo and determines if they'll ever land at the same location at the same time?

- Write a python program to read the text file lotto.txt that has one integer per line and print the lowest non-repeating integer among all the integers in the file.

- Write a python program to compute decimal value of a given string if it is a valid Roman numeral and outputs "Not a valid Roman Numeral" if it is not.

  Here are some Roman numeral strings and their associated values:

  'I' 1 'II' 2 'MCMLXXXIV' 1984

  'V' 5 'IV' 4 'MMMDCCCLXXXVIII' 3888

  'X' 10 'IX' 9 'CDIX' 409

  'L' 50 'XL' 40 'MMCXL' 2140

  'C' 100 'XC' 90 'CI' 101

  'D' 500 'CD' 400 'MLI' 1051

  'M' 1000 'CM' 900 'CCCXL' 340

  There are three properties that need to be satisfied:

  1) There can be at most three X's

  2) None of the characters in LXVI can come before an M

  3) There can be at most one CM.

  4) Only characters M, D, C, L, X, V, and I are legal

  Sometimes C, X, and I indicate negative values.

  For example, the value of CMXCIX is -100 + 1000 - 10 + 100 -1 + 10 = 999.

Projects

- Write a python program to find and print the maximum level of nesting in a given a valid XML document. The file name will be given as input to this program and it will prints the depth of the file (Take the depth of the root as 0)

- Write a python program to extract all valid email addresses from a text file (emails.txt)

  A valid email address meets the following criteria:

  It's composed of a username, domain name, and extension assembled in this format:

  username@domain.extension

  The username starts with an English alphabetical character, and any subsequent characters consist of one or more of the following: alphanumeric characters, -,., and _.

  The domain and extension contain only English alphabetical characters.

  The extension is 1, 2, or 3 characters in length.

- Write a python program to validate national id unique identification number (UID) for all the citizens of a nation. The program will read a text file (citizens.txt) that has a number of UIDs and prints which are valid and which are not valid

  A valid UID must follow the rules below:

  - It must contain at least 2 uppercase English alphabet characters.
  - It must contain at least 3 digits
  - It should only contain alphanumeric characters
  - No character should repeat.
  - There must be exactly 10 characters in a valid UID.

Projects

When you shuffle cards you make two stacks, then interleave them, and if you were to do it perfectly, the even numbered cards would come from one stack and the odd numbered cards from the other stack. We call this even-odd interleaving a "perfect shuffle." Of course, you can do the same thing with any two lists, regardless of what type they contain.
Write python program with a recursive function that shuffles together two lists in this way. Here is the specification for the recursive function:

def shuffle(a, b):
"""Perfectly shuffle two decks. Return a new list that contains the items
in a and b, interleaved in the order a[0], b[0], a[1], b[1], a[2], ...
If one list is longer than the other, the extra items go at the end.
"""

A matrix can be represented using a list of sublists of equal lengths, with each sublist representing a row in the matrix. For example, the matrix

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 0 & 5 & 0 \\ 6 & 7 & 8 & 9 \end{bmatrix}$$

can be represented as:

```
M0 = [ [1, 2, 3, 4],  [5, 0, 5, 0], [6, 7, 8, 9]]
```

Write a python function with the signature `largest_col_sum(M)` which takes in a list of lists M which represents a matrix in the format outlined above, and returns the sum of the column in the matrix whose sum is the largest. For example, in the matrix `M0` above, the third column has the largest sum: $(3 + 5 + 8 = 16)$. So `largest_col_sum(M0)` should return `16`.

Projects

- Without using `for-` or `while-` loops, write a function with the signature `filter_out_odds(L)` which takes in a list `L` of integers, and returns a new list that contains only the even integers in `L`, in the same order as in `L`. For example, `filter_out_odds([5, -2, 4, 0, 3, 7, 8])` should return `[-2, 4, 0, 8]`.

- Write a function with the signature `ev(expr)` that takes in a string containing an expression of the form `"<int_1><operation_1><int_2><operation_2><int_3><operation_3>...<int_k>"` and returns the value of that expression. For example,

  `ev("10")` should return 10

  `ev("10+2")` should return 12

  `ev("14-3*2")` should return 8

  Assume `expr` is a valid non-empty expression. Assume `<int_1>`, `<int_2>`, ..., `<int_k>` can only be positive integers, and the operations can only be `"+"`, `"-"`, or `"*"` (there is no division).

Projects

- We will call a dictionary *cyclic* if there is at least one sequence of key-value pairs such that the value of one pair matches the key of the next pair, and the value of the last pair matches the key of the first pair. For instance, consider the following dictionary:

```
example_d = { 'awesome': 'bodacious', 'soda': 'cs',
              'bodacious': 'soda', 'cs': 'awesome', '61a': 'cool'
```

This dictionary is cyclic because we have the chain:

```
'awesome' -> 'bodacious' -> 'soda' -> 'cs' -> 'awesome'
```

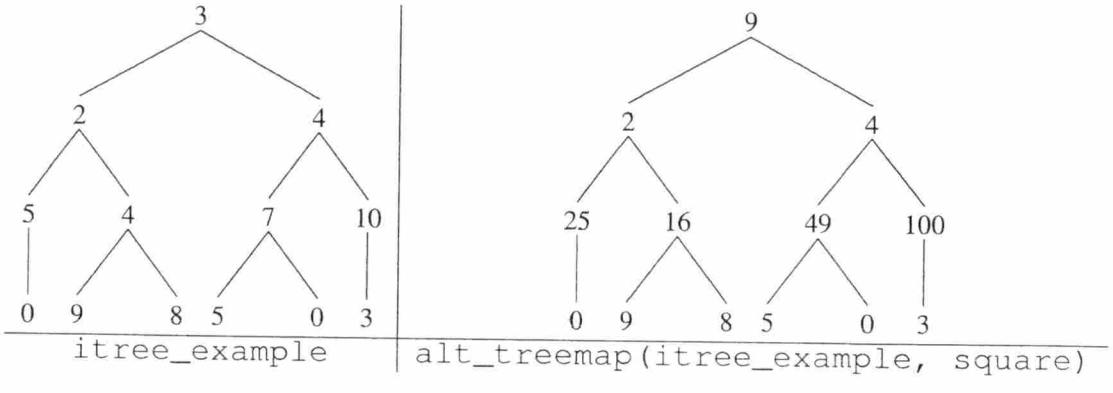
where `awesome` maps to `bodacious`, `bodacious` maps to `soda`, and so on, until we find `awesome` again. Note that the entire dictionary does **not** need to be a cycle. In the dictionary `example_d`, for instance, there is a key-value pair that is not part of the cycle: `('61a', 'cool')`. The dictionary is still considered cyclic because at least one cycle exists.

Define the procedure `is_cyclic` that, given a dictionary `dict`, returns `True` if `dict` is cyclic.

Hint: It may be helpful to write a helper that takes a starting key and checks if there is a cycle starting with that key.

- Write the function `alt_treemap` that, given a function and an ITree, applies the function to all of the data at every other level of the ITree, starting at the root node.

  For example, given the ITree on the left (`itree_example`) and the `square` function, the output of the function should be the ITree on the right:



You may not convert the ITree to any other different data structure (such as tuples, lists, or other sequences)

Projects

- We have a sequence of numeric data points `seq1`, and we want to see if a sequence of relevant numbers `seq2` is found in the data. The catch is, we want to see if the numbers in `seq2` occur *in the same order* within the data of `seq1`, though not necessarily one after the other. If so, then `seq2` is a *subsequence* of `seq1`.

  Write a predicate function `is_subseq` that takes two tuples `seq1` and `seq2` as arguments, and determines if `seq2` is a subsequence of `seq1`. If so, the function should return `True`; otherwise, it should return `False`.

  Do not use a `while`-loop or `for`-loop in your solution.

```
def is_subseq(seq1, seq2):
    """Returns True if seq2 is a subsequence of seq1.

    >>> is_subseq((9, 1, 4, 5, 6), (4, 5, 6))
    True
    >>> is_subseq((3, 5, 0, 3, 4, 3, 7, 9, 3, 2), (3, 3, 9, 2))
    True
    >>> # Below, the numbers in seq2 appear in seq1,
    >>> # but not in the same order.
    >>> is_subseq((3, 5, 5, 8, 3), (8, 5, 3))
    False
    >>> # Below, not all the numbers in seq2 are present in seq1.
    >>> is_subseq((3, 5, 5, 8, 3), (3, 2, 8))
    False
    >>> is_subseq((3, 2, 57, 8), (3, 5, 7))
    False
    """
```

Projects

- Write a python function `difference` that takes two dictionaries `d1`, `d2` and that returns a third dictionary `d` such that `d` contains every key that appears in both `d1` and `d2` but with different values (*i.e.*, where the values `d1[key]` and `d2[key]` are different). For each such `key`, you should set the value of `d[key]` to the tuple `(d1[key], d2[key])`. For example, if

```
d1 = { "a": 1, "b": 2, "c": 3 }
d2 = { "b": 2, "c": 4, "d": 6 }
```

  then `difference(d1, d2)` returns `{ "c": (3,4) }`.

- Write a function `flatten` that takes a list `L` and that returns a list of all non-list elements nested within `L`, no matter how many sub-lists they are nested within. For example,

```
flatten([[]])                 returns []
flatten([1])                  returns [1]
flatten([[2,3]])              returns [2,3]
flatten([[1,[]],[[[2]],3]]) returns [1,2,3]
```

- In HTML, all text that appears between the tags `<b>` and `</b>` is displayed in **bold**. Write a python program with a function `get_bold_text` that takes the URL of a web page (as a string) and that returns all of the bold text on that page (as a list of strings). For example, if the page at `http://url1.html` contains

```
<html><i>Hello</i> there!</html>
```

  and the page at `http://url2.html` contains

```
<html>
This is a <b>simple</b> web page, with <B>NOTHING</b> complicated on it!
</html>
```

  then

```
get_bold_text("http://url1.html") returns []
get_bold_text("http://url2.html") returns ["simple", "NOTHING"]
```

  Remember that HTML tags are case-*insensitive*, *i.e.*, they can be uppercase (`<B>`, `</B>`) or lowercase (`<b>`, `</b>`) and the case of the closing tag does not have to match that of the opening tag—as in the second example above. Assume that the web page uses properly formatted HTML (*i.e.*, all open tags are closed) and that there are **no** nested bold tags (*i.e.* **nothing** like `"<b>bold <b>bolder?</b> bold</b>"`).

Projects

- We want to compress strings that have long sequences of equal characters. For example, we want to compress `'bbbbaaa$$$$$$$$$$$$$$$d'` to `'b4a3$16d1'`. In the compression, each sequence of equal characters is given by the character followed by the length of the sequence. Write the function `compress` to do this. Use no loops; use only recursion. You may use the function `eq chars` specifed below as a helper.

  Hint: The base case is not necessarily a string with one character

- A triangular array is a ragged list whose rows start (end) at size one and increase (decrease) by one each row. Examples of triangular arrays are shown to the right. These arrays have applications in scientific computation.

  The transpose is a triangular array with rows and columns swapped with each other. The two triangular arrays shown to the right are the transpose of one another.

  Write the function `transpose`, which takes an increasing triangular array and returns the (decreasing) transpose of the original.

1	
2 0	
5 6 -1	
4 -2 3 8	

Triangular, Increasing

1 2 5 4	
0 6 -2	
-1 3	
8	

Triangular, Decreasing

- RGB objects are with three attributes – `red`, `green`, and `blue`

  which all had the invariant that they must be an `int` between 0 and 255 (inclusive). The constructor for an RGB object is the function `RGB(r,g,b)` where `r`,`g`,`b` are the attribute values

  Greyscale is an integer between 0 and 255, given by the formula

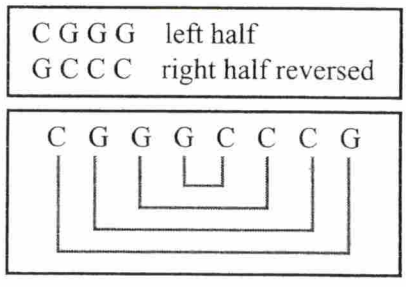
  $$\texttt{grey} = 0.21\texttt{R} + 0.71\texttt{G} + 0.07\texttt{B}$$

  We often store grey colors as an RGB object where `red`, `green`, and `blue` are all equal (to the grey value). Write a python program to implement the function rgb_to_grey

  ```
  def rgb_to_grey(rgb):
  ```

Projects

m-RNA (messenger RNA) is a chain or string of "nucleotides." These nucleotides are typically denoted by the four symbols `'C'` (cytosine), `'G'` (guanine), `'A'` (adenine), and `'U'` (uracil). When writing programs to process m-RNA, we typically represent them as a string of these four letters. For example, `'AUUGC'` is an m-RNA sequence.



Each nucleotide has a complementary nucleotide with which it bonds; `'C'` **and** `'G'` **are a complementary pair, as are** `'A'` **and** `'U'`. We call the RNA sequence `'CGGGCCCG'` a perfect hinge because, if the right half is reversed and placed under the left half, the corresponding characters are complementary. This is shown above. From this definition of a perfect hinge, we can deduce that **an empty sequence is a perfect hinge and a sequence with an odd number of elements is not a perfect hinge.**

Structures similar to perfect hinges play an important role in biology. Hence, we often want to detect whether an m-RNA sequence has this property. Implement the function below with the given specification. Use recursion, not a loop.

```
def ishinge(seq):
    """Returns: True if the m-RNA sequence is a perfect hinge.  Returns
    False otherwise (e.g. not even, does not match).

    Precondition: seq is a string with characters 'C', 'G', 'A', 'U'."""
```

Projects

- Write a python program that prints out all the possible 4-letter strings that can be written using the letters "a", "b", "c", "d", "e", "f" and "g" (and no other character) such that no letter appears in any string more than twice. For example, `"ccbb"` should be printed, but `"aaab"` should not. Hint: think about how you would print *all* the 4-letter strings.

- Write a function with the signature `near_anagram(w1, w2)` that takes in two strings, `w1` and `w2`, and returns `True` iff `w1` can be obtained from `w2` by rearranging the characters in `w2`, and then changing exactly one of the characters for another character. For example `near_anagram("cat", "tap")` is `True` since `"cat"` can be rearranged into `"tac"`, and then changed into `"tap"` by exchanging `c` for `p`. Assume `w1` and `w2` only contain lowercase letters.

- A list is symmetric if it is the same read from the left and from the right. For example, `[1, 2, 3, 2, 1]` is symmetric, `[1, 2, 2, 1]` is symmetric, but `[1, 2, 3]` is not symmetric. Write the function `is_symmetric(L)` that returns `True` iff the list `L` is symmetric.

- Write a function that takes in a list of 8 integers, and returns `True` iff some non-empty subset of the integers sums up to 0. For example, the function should return `True` for `[-3, 10, 20, 5, 10, 11, 2, 1]` since $-3 + 2 + 1 = 0$, and the function should return `False` for `[1, 2, 3, 4, 5, 6, 7, 8]`, since no subset of those numbers sums up to 0.

Projects

- Write a python program with the following function so that it performs as specified.

```
def OverBudget(A,M):
 """ Returns the smallest k so that sum(abs(A[0:k,0]))>=M, sum(abs(A[0:k,1]))>=M, and
 sum(abs(A[0:k,2]))>=M. If no such k exists, returns 0.
 PreC: A is an n-by-3 numpy array of ints. M is an int.
 """
```

To illustrate, suppose

$$A = \begin{bmatrix} 2 & 7 & 1 \\ 1 & 0 & 4 \\ 3 & 2 & 5 \\ 0 & 1 & 4 \\ 4 & 0 & 6 \end{bmatrix}$$

If `M = 3`, then the value returned should be 2. If `M = 10`, then the returned value should be 5. If `M = 100`, then the returned value should be 0. Do not to use the built-in function `sum` or `for`-loops.

- Write a function with the signature `is_sorted(L)` which takes in a list of `ints` `L`, and returns `True` iff the list `L` is sorted, in either non-increasing or non-decreasing order. For example,

`is_sorted([4, 5, 6, 1, 2, 3, 7])` should return `False`, `is_sorted([4, 5, 5, 6])` should return `True`, `is_sorted([6, 3])` should return `True`, `is_sorted([])` should return `True`.

Projects

- Evaluate the dataset containing the GDPs of different countries from the file 'GDP.txt' to:
  - Find and print the name of the country with the highest GDP,
  - Find and print the name of the country with the lowest GDP,
  - Print out text and input values iteratively,
  - Print out the entire list of the countries with their GDPs, and
  - Print the highest GDP value, lowest GDP value, mean GDP value, standardized GDP value, and the sum of all the GDPs.

- Evaluate the dataset of the Summer Olympics, London 2012 from the file 'olympics_medal_tally.xls' to:
  - Find and print the name of the country that won maximum gold medals,
  - Find and print the countries who won more than 20 gold medals,
  - Print the medal tally,
  - Print each country name with the corresponding number of gold medals, and
  - Print each country name with the total number of medals won.

- Use SciPy to declare 20 random values for random values and perform the following:
  1. CDF – Cumulative Distribution Function for 10 random variables.
  2. PDF – Probability Density Function for 14 random variables.

- Analyze the Federal Aviation Authority (FAA) dataset using Pandas to do the following:
  1. View
     a. aircraft make name
     b. state name
     c. aircraft model name
     d. text information
     e. flight phase
     f. event description type
     g. fatal flag
  2. Clean the dataset and replace the fatal flag NaN with "No"
  3. Find the aircraft types and their occurrences in the dataset
  4. Remove all the observations where aircraft names are not available
  5. Display the observations where fatal flag is "Yes"

Projects

- Write a python function with the signature `movies_by_release_date(movies)` which takes in a dictionary whose keys are movie names and whose values are release dates, and which returns a list of movie names, in order from the most recent release date to the earliest release date. The release dates are either in the format `"<year>, in <location>"`, or in the format `"a long time ago, in <location>"`. Any movies released `"a long time ago"` were released before the movies for which the year is indicated. Movies released at the same time can be placed in the list in any order.

  For example, if `movies` equals

```
{"Dude, Where's My Death Star": "a long time ago, in a galaxy far far away",
 "Star Wars: The Force Awakens": "2015, in Los Angeles",
 "Star Wars": "1977, in Los Angeles",
 "Sleepless in Aldera":  "a long time ago, in Alderaan City",
 "Jurassic World": "2015, in New York"},
```

  `movies_by_release_year(movies)` will return

```
["Jurassic World", "Star Wars: The Force Awakens", "Star Wars",
 "Sleepless in Aldera", "Dude, Where's My Death Star" ].
```

- Write a recursive function with the signature `merge(L1, L2)` which takes in two lists sorted in non-decreasing order, and returns a list that contains all the elements from both `L1` and `L2`, and is itself sorted. **You may not use loops, global variables, or helper functions, and the function signature must be exactly as specified (i.e., you may not add additional parameters).** You may not use Python's `sorted()` and `sort()` functions. You may use slicing.

  For example, `merge([4, 8, 10], [2, 5])` should return `[2, 4, 5, 8, 10]`.

Projects

- FDNY dataset in CSV format is given for the Fire Department of New York City. Analyze the dataset to determine:
  1. The total number of fire department facilities in New York city
  2. The number of fire department facilities in each borough
  3. The facility names in Manhattan

- The given dataset contains ad budgets for different media channels and the corresponding ad sales of Nyka Inc. Evaluate the dataset to:
  - Find the features or media channels used by the firm
  - Find the sales figures for each channel
  - Create a model to predict the sales outcome
  - Split as training and testing datasets for the model
  - Calculate the Mean Square Error (MSE)

- The pima_indians_diabetes.data lists the glucose level readings of several pregnant women taken either during a survey examination or routine medical care. It specifies if the 2 hour post-load plasma glucose was at least 200 mg/dl. Analyze the dataset to:
  1. Find the features of the dataset,
  2. Find the response label of the dataset,
  3. Create a model to predict the diabetes outcome,
  4. Use training and testing datasets to train the model, and
  5. Check the accuracy of the model.

- Analyze the given Spam Collection dataset to:
  1. View information on the spam data,
  2. View the length of messages,
  3. Define a function to eliminate stopwords,
  4. Apply Bag of Words,
  5. Apply tf-idf transformer, and
  6. Detect Spam with Naïve Bayes model.

Projects

- Analyze the Sentiment dataset using NLP to:
  1. View the observations,
  2. Verify the length of the messages and add it as a new column,
  3. Apply a transformer and fit the data in the bag of words,
  4. Print the shape for the transformer, and
  5. Check the model for predicted and expected values.

- Using matplotlib and seaborn libraries, analyze the "auto mpg data" and draw a pair plot for mpg, weight, and origin.

  This dataset was taken from the StatLib library maintained at Carnegie Mellon University.

  - mpg: continuous
  - cylinders: multi-valued discrete
  - displacement: continuous
  - horsepower: continuous

- You have been provided with a dataset that lists Ohio State's leading causes of death.

  Using the two data points:

  - Cause of deaths and
  - Percentile

  Draw appropriate visualizations to visualize the dataset.

Projects

- One of the leading retail stores in the US, Walmart, would like to predict the sales and demand accurately. There are certain events and holidays which impact sales on each day. There are sales data available for 45 stores of Walmart. The business is facing a challenge due to unforeseen demands and runs out of stock some times, due to the inappropriate machine learning algorithm. An ideal ML algorithm will predict demand accurately and ingest factors like economic conditions including CPI, Unemployment Index, etc.

  Walmart runs several promotional markdown events throughout the year. These markdowns precede prominent holidays, the four largest of all, which are the Super Bowl, Labor Day, Thanksgiving, and Christmas. The weeks including these holidays are weighted five times higher in the evaluation than non-holiday weeks. Part of the challenge presented by this competition is modeling the effects of markdowns on these holiday weeks in the absence of complete/ideal historical data. Historical sales data for 45 Walmart stores located in different regions are available. Using the data answer the following:

  - Which store has maximum sales
  - Which store has maximum standard deviation i.e., the sales vary a lot. Also, find out the coefficient of mean to standard deviation
  - Which store/s has good quarterly growth rate in Q3'2012
  - Some holidays have a negative impact on sales. Find out holidays which have higher sales than the mean sales in non-holiday season for all stores together
  - Provide a monthly and semester view of sales in units and give insights

  For Store 1 – Build prediction models to forecast demand

  - Linear Regression – Utilize variables like date and restructure dates as 1 for 5 Feb 2010 (starting from the earliest date in order).
  - Hypothesize if CPI, unemployment, and fuel price have any impact on sales.

Projects

A popular mobile phone brand, Lenovo has launched their budget smartphone in the Indian market. The client wants to understand the VOC (voice of the customer) on the product. This will be useful to not just evaluate the current product, but to also get some direction for developing the product pipeline. The client is particularly interested in the different aspects that customers care about. Product reviews by customers on a leading e-commerce site should provide a good view.

Discover the topics in the reviews and present it to business in a consumable format.
Employ techniques in syntactic processing and topic modeling.
Perform specific cleanup, POS tagging, and restricting to relevant POS tags, then, perform topic modeling using LDA. Finally, give business-friendly names to the topics and make a table for business.

1. Normalize casings for the review text and extract the text into a list for easier manipulation.
2. Tokenize the reviews using NLTKs word_tokenize function.
3. Perform parts-of-speech tagging on each sentence using the NLTK POS tagger.
4. For the topic model, we should want to include only nouns.
   a. Find out all the POS tags that correspond to nouns.
   b. Limit the data to only terms with these tags.
5. Lemmatize.
   a. Different forms of the terms need to be treated as one.
   b. No need to provide POS tag to lemmatizer for now.
6. Remove stopwords and punctuation .

Projects

- Twitter is the biggest platform where anybody and everybody can have their views heard. Some of these voices spread hate and negativity. Twitter is wary of its platform being used as a medium to spread hate.

  You are a data scientist at Twitter, and you will help Twitter in identifying the tweets with hate speech and removing them from the platform. You will use NLP techniques, perform specific cleanup for tweets data, and make a robust model.

  - Normalize the casing.
  - Using regular expressions, remove user handles. These begin with '@'.
  - Using regular expressions, remove URLs.
  - Remove stop words.
  - Remove redundant terms like 'amp', 'rt', etc.
  - Remove '#' symbols from the tweet while retaining the term.

Projects

- Zomato is India's largest platform for discovering restaurants and ordering food. It operates in India as well as a few cities internationally. Bangalore is one of the biggest customer and restaurant bases for Zomato with 4 to 5 million users using the platform each month.

  Users on the platform can also post reviews of restaurants, and provide a rating accompanying the review. The content in the reviews should ideally reflect the rating provided by the customer. In many cases, there is a mismatch, owing to multiple reasons where the rating does not match the customer review. The reviews and ratings matching is very important as it builds customer trust on the platform, and helps the user get an accurate picture of the restaurant.

  You, as a data scientist, need to enable the identification and cleanup of such cases, to ensure the ratings are reflective of the reviews and that the reviews seem trustworthy to the customer. You will need to use NLP techniques in conjunction with Machine learning models to predict the rating from the review text.

  1. Load the data using read_csv function from pandas package
  2. Null values in the review text?
  3. Perform cleanup on the data
  4. Model building: Random Forest Regressor
  5. Model evaluation
  6. Hyperparameter tuning
  7. Find the parameters with the best mean square error in cross validation

Projects

- A car showroom owner would like to develop a model to predict the total dollar amount that customers are willing to pay given the following attributes:
  - Customer Name
  - Customer e-mail
  - Country
  - Gender
  - Age
  - Annual Salary
  - Credit Card Debt
  - Net Worth

- Build an image classifier using CIFAR-10 is a dataset that consists of several images divided into the following 10 classes:

  Airplanes, Cars, Birds, Cats, Deer, Dogs, Frogs, Horses, Ships, Trucks

  The dataset stands for the Canadian Institute For Advanced Research (CIFAR)

  CIFAR-10 is widely used for machine learning and computer vision applications.

  The dataset consists of 60,000 32x32 color images, 6,000 images of each class.

  Images have low resolution (32x32).

- Build an machine learning model to predict price of avocado sold in the US using historical dataset.

  Data represents weekly 2018 retail scan data for National retail volume (units) and price.

  The data is as follows:

  Date - The date of the observation

  Average Price - the average price of a single avocado

  type - conventional or organic

  year - the year

  Region - the city or region of the observation

  Total Volume - Total number of avocados sold

Projects

- Build a deep learning model for Traffic sign classification. Traffic sign classification is an important task for self driving cars.

  The dataset contains 43 different classes of images.

  Classes are as listed below:

  (0, b'Speed limit (20km/h)')
  (1, b'Speed limit (30km/h)')
  (2, b'Speed limit (50km/h)')
  (3, b'Speed limit (60km/h)')
  (4, b'Speed limit (70km/h)')
  (5, b'Speed limit (80km/h)')
  (6, b'End of speed limit (80km/h)')
  (7, b'Speed limit (100km/h)')
  (8, b'Speed limit (120km/h)')
  (9, b'No passing')
  (10, b'No passing for vehicles over 3.5 metric tons')
  (11, b'Right-of-way at the next intersection')
  (12, b'Priority road')
  (13, b'Yield')
  (14, b'Stop')
  (15, b'No vehicles')
  (16, b'Vehicles over 3.5 metric tons prohibited')
  (17, b'No entry')
  (18, b'General caution')
  (19, b'Dangerous curve to the left')
  (20, b'Dangerous curve to the right')
  (21, b'Double curve')
  (22, b'Bumpy road')
  (23, b'Slippery road')
  (24, b'Road narrows on the right')
  (25, b'Road work')
  (26, b'Traffic signals')
  (27, b'Pedestrians')
  (28, b'Children crossing')
  (29, b'Bicycles crossing')
  (30, b'Beware of ice/snow')
  (31, b'Wild animals crossing')
  (32, b'End of all speed and passing limits')
  (33, b'Turn right ahead')
  (34, b'Turn left ahead')
  (35, b'Ahead only')
  (36, b'Go straight or right')
  (37, b'Go straight or left')
  (38, b'Keep right')
  (39, b'Keep left')
  (40, b'Roundabout mandatory')
  (41, b'End of no passing')
  (42, b'End of no passing by vehicles over 3.5 metric tons')

Projects

- Build a classification model for classifying SMSs into spam (unwanted messages) or ham (good messages)
  The SMS Spam Collection is a set of SMS tagged messages that have been collected for SMS Spam research.
  It contains one set of SMS messages in English of 5,574 messages, tagged according being ham (legitimate) or spam.
  The files contain one message per line. Each line is composed by two columns: v1 contains the label (ham or spam) and v2 contains the raw text.

- Build a deep learning model using TensorFlow2.x to predict bike rental usage based on inputs such as temperature, humidity, wind speed etc. Data points include:
  - instant: record index
  - dteday: date
  - season: season (1: springer, 2: summer, 3: fall, 4: winter)
  - yr: year (0: 2011, 1: 2012)
  - mnth: month (1 to 12)
  - hr: hour (0 to 23)
  - holiday: whether day is holiday or not - weekday : day of the week
  - Working day: if day is neither weekend nor holiday is 1, otherwise is 0.

- Build a deep learning model using TensorFlow2.x to predict house sale prices for King County in Washington, USA.
  Homes that are sold in the time period: May, 2014 and May, 2015.
  Model inputs:
  - ida: notation for a house
  - date: Date house was sold
  - bedrooms: Number of Bedrooms
  - bathrooms: Number of bathrooms
  - sqft_living: home square footage
  - sqft_lot: square footage of the lot
  - floors: Total floors (levels) in house
  - waterfront: waterfront property

Projects

- Build a deep learning model using TensorFlow2.x to perform sentiment analysis of Amazon customer reviews.

  Dataset consists of 3000 Amazon customer reviews, star ratings, date of review, variant and feedback of various amazon Alexa products like Alexa Echo, Echo dots.

  The objective is to discover insights into consumer reviews and perform sentiment analysis on the data.

- Build a deep learning model using TensorFlow2.x to perform fashion accessory classification using Fashion dataset consisting of 70,000 images in 10 categories:
  - T-SHIRT/TOP
  - TROUSER
  - PULLOVER
  - DRESS
  - COAT
  - SANDAL
  - SHIRT
  - SNEAKER
  - BAG
  - ANKLE BOOT

Projects

- Perform EDA on the insurance data. In the case of an insurance company, attributes of customers like age, BMI etc can be crucial in making business decisions.
  1. age: age of primary beneficiary
  2. sex: insurance contractor gender, female, male
  3. bmi: Body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9
  4. children: Number of children covered by health insurance / Number of dependents
  5. smoker: Smoking
  6. region: the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.
  7. charges: Individual medical costs billed by health insurance.

  EDA tasks include:

  a. Finding the shape of the data
  b. Data type of each attribute
  c. Checking the presence of missing values
  d. 5 point summary of numerical attributes
  e. Distribution of 'bmi', 'age' and 'charges' columns.
  f. Measure of skewness of 'bmi', 'age' and 'charges' columns
  g. Checking the presence of outliers in 'bmi', 'age' and 'charges columns
  h. Distribution of categorical columns (include children)
  i. Pair plot that includes all the columns of the data frame

Projects

- Data analysis of La Liga team. La Ligais the men's top professional football division of the Spanish football league system. The dataset contains information on all the teams that have participated in all the past tournaments. It has data about how many goals each team scored, conceded, how many times they came within the first 6 positions, how many seasons they have qualified, their best position in the past, etc.
  - Pos - Position in among the list of all teams
  - Team Seasons - how many seasons team has played so far
  - Points - total number of points of the team
  - GamesPlayed - total number of games played so far
  - GamesWon - total number of games won so far
  - GamesDrawn - total number of games drawn so far
  - GamesLost - total number of games lost so far
  - GoalsFor - total number of goals by the team
  - GoalsAgainst - total number of goals against the team
  - Champion - total number of times it team is a champion
  - Runner-up - total number of times it team is a runner-up
  - Debut - debut year
  - BestPosition - best position of the team

EDA Tasks:

a. Print all the teams which have started playing between 1930-1980.
b. Print the list of teams which came Top 5 in terms of points
c. Write a function with the name "Goal_diff_count" which should return all the teams with their Goal Differences.
d. Using the same function, find the team which has the maximum and minimum goal difference.
e. Create a new column with the name "Winning Percent" and append it to the data set
f. Print the top 5 teams which have the highest Winning percentage
g. Group teams based on their "Best position" and print the sum of their points for all positions

Projects

- Build a Machine Learning Model to predict the condition of the patient depending on the test results. The data consists of biomechanics features of the patients according to their current conditions. Each patient is represented in the data set by six biomechanics attributes derived from the shape and orientation of the condition to their body part.

  1. P_incidence
  2. P_tilt
  3. L_angle
  4. S_slope
  5. P_radius
  6. S_degree
  7. Class

  Tasks:

  a. Perform detailed statistical analysis on the data.
  b. Perform a detailed univariate, bivariate and multivariate analysis with appropriate detailed comments after each analysis.
  c. Segregate predictors vs target attributes
  d. Perform normalisation or scaling if required.
  e. Design and train a KNN classifier.
  f. Display the classification accuracies for train and test data.
  g. Display and explain the classification report in detail.
  h. Automate the task of finding best values of K for KNN.
  i. Apply all the possible tuning techniques to train the best model for the given data. Select the final best trained model with your comments for selecting this model.

Projects

- Build a machine learning model to perform focused marketing by predicting the potential customers who will convert using the historical dataset of First Gibralter Bank of Texas. The bank is on a massive digital transformation for all its departments. It has a growing customer base with majority of them are liability customers (depositors) vs borrowers (asset customers). The bank is interested in expanding the borrowers base rapidly to bring in more business via loan interests. A campaign that the bank ran in last quarter showed an average single digit conversion rate. Digital transformation being the core strength of the business strategy, marketing department wants to devise effective campaigns with better target marketing to increase the conversion ratio to double digit with same budget as per last campaign.

  The data set has the following attributes:

  1. ID: Customer ID
  2. Age Customer's approximate age.
  3. CustomerSince: Customer of the bank since. [unit is masked]
  4. HighestSpend: Customer's highest spend so far in one transaction. [unit is masked]
  5. ZipCode: Customer's zip code.
  6. HiddenScore: A score associated to the customer which is masked by the bank as an IP.
  7. MonthlyAverageSpend: Customer's monthly average spend so far. [unit is masked]
  8. Level: A level associated to the customer which is masked by the bank as an IP.
  9. Mortgage: Customer's mortgage. [unit is masked]
  10. Security: Customer's security asset with the bank. [unit is masked]
  11. FixedDepositAccount: Customer's fixed deposit account with the bank. [unit is masked]
  12. InternetBanking: if the customer uses internet banking.
  13. CreditCard: if the customer uses bank's credit card.
  14. LoanOnCard: if the customer has a loan on credit card.

  Tasks:

  - Design and train a Logistic regression and Naive Bayes classifiers.
  - Display the classification accuracies for train and test data.
  - Display and explain the classification report in detail.
  - Apply all the possible tuning techniques to train the best model for the given data. Select the final best trained model with comments for selecting this model.