## Project 6: Donald Duck's Digital Data Ducts

### A. Introduction

==========

From: Mickey Mouse (CEO)
To: All Employees
Cc: All Company Executives
Date: 10:15 a.m., October 22
Subject: Dissolving Mickey Mouse Clubhouse

To fellow employees of Mickey Mouse Clubhouse,

We regret to inform you that we will be closing the Mickey Mouse Clubhouse as of (date). The general business decline has made it impossible to justify keeping the company open to the relevant stakeholders. To give you some context, as we all know during the COVID situation, the closing of our amusement parks has largely reduced our cash flow, combining with upcoming competition from alternative streaming platforms such as YouTube and TikTok videos which has captured a significant portion of our market share.

We thank you and wish you the best of success in your new positions, wherever it may be.

Mickey Mouse,
CEO, Mickey Mouse Clubhouse

==========

And that is how the once lovable pantsless entertainer, became jobless. Desperate to reclaim his former glory in the social ladder, he decides he's going to take up the coolest job of the decade, Data Scientist (debatable, chicken rice sellers are pretty much elite class in the UM hierarchy). But he is new to programming so he can't do it from scratch. He needs a library. And you'll be the ones to build it for him.

### B. Problem Statement

You are required to develop a library to help Donald on his Data Science journey. Inputs and outputs will be in CSV format. CSV stands for comma separated values, meaning the values are separated by a comma as delimiter. Each line is a row and each value in the row is a column value. Sound familiar? It is basically Excel. In fact, opening CSV files on a windows computer will open it in excel. The first row of the CSV file usually contains the column labels.

Example CSV:

Name,Department,CurrentCGPA,Expected Graduation salary,Actual graduation salary

Meow,Artificial Intelligence,3.7,1000000,1000

Woof,Software Engineering,2.0,4200,4200

LWY,Information Systems,4.3,1000,1000000

Excel equivalent:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Name | Department | CurrentCGPA | Expected Graduation salary | Actual graduation salary |
| 2 | Meow | Artificial Intelligence | 3.7 | 1000000 | 1000 |
| 3 | Woof | Software Engineering | 3 | 4200 | 4200 |
| 4 | LWY | Information Systems | 4.3 | 1000 | 1000000 |
| 5 | | | | | |

The actual CSV specification is a bit more complicated than this as it is possible for a value to be a string containing commas which would ruin the strategy of simply splitting by comma. **However, to keep this question fun, you can assume that splitting by commas is enough to tokenize the line in the context of this assignment. In the spirit of having fun, you are also only required to parse integers, floats and strings.**

The library needs to contain:

1. DataFrame Object

   A DataFrame object to interface with the CSV file parsed.

   **a. Method to save DataFrame to CSV file**

   **b. Method to construct DataFrame from CSV file**

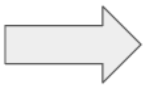
2. Manipulation methods

   **a. Method to concatenate DataFrames.**

   *Column concatenation means stacking columns.*

| Fruit | Quantity |
|---|---|
| Potato | 420 |
| Tomato | 404 |
| Plato | 1 |

| Color |
|---|
| Potato Color |
| Red |
| Platato Color |



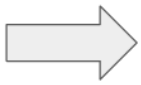
| Fruit | Quantity | Color |
|---|---|---|
| Potato | 420 | Potato Color |
| Tomato | 404 | Red |
| Plato | 1 | Platato Color |

   *Row concatenation means stacking rows.*

| Fruit | Quantity | Color |
|---|---|---|
| Potato | 420 | Potato Color |
| Tomato | 404 | Red |

+

| Fruit | Quantity | Color |
|---|---|---|
| Plato | 1 | Platato Color |



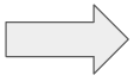
| Fruit | Quantity | Color |
|---|---|---|
| Potato | 420 | Potato Color |
| Tomato | 404 | Red |
| Plato | 1 | Platato Color |

*If the combining axis doesn't match, the program should provide an error to tell Donald he made a mistake.*

| Fruit | Quantity | Color |
|---|---|---|
| Potato | 420 | Potato Color |
| Tomato | 404 | Red |

+

| Fruit | Quantity | Color | Dead |
|---|---|---|---|
| Plato | 1 | Platato Color | Yes |



Unequal number of columns

**b. Method to obtain a subset of DataFrame with range of row or column.** Rows are 0 indexed and the range is inclusive of the first element but exclusive of the last.

*DataInFirstExample.rowRange(1,3)*

| Name | Department | CurrentCGPA | Expected Graduation | Actual graduation salary |
|---|---|---|---|---|
| Woof | Software Engineering | 2 | 4200 | 4200 |
| LWY | Information Systems | 4.3 | 1000 | 1000000 |

*DataInFirstExample.colRange(new String[] {"Department","Actual graduation salary"})*

| Department | Actual graduation salary |
|---|---|
| Software Engineering | 4200 |
| Artificial Intelligence | 1000 |
| Information Systems | 1000000 |

**c. Method to sort the rows by a column in the DataFrame.**

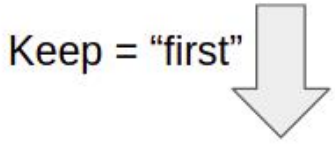
*DataInFirstExample.sort("CurrentCGPA")*

| Name | Department | CurrentCGPA | Expected Graduation | Actual graduation salary |
|---|---|---|---|---|
| Woof | Software Engineering | 2 | 4200 | 4200 |
| Meow | Artificial Intelligence | 3.7 | 1000000 | 1000 |
| LWY | Information Systems | 4.3 | 1000 | 1000000 |

**d. Method to remove duplicate rows based on subset of columns. There should be a parameter to choose whether to keep the first, last, specific number or no occurrence.**

*Data.dropDuplicates(new String[] {“Name”},”first”)*

| Name | Department | CurrentCGPA | Expected Graduation | Actual graduation salary |
|---|---|---|---|---|
| Emily | Software Engineering | 2 | 4200 | 4200 |
| Emily | Artificial Intelligence | 3.7 | 1000000 | 1000 |
| Boomily | Information Systems | 4.3 | 1000 | 1000000 |




| Name | Department | CurrentCGPA | Expected Graduation | Actual graduation salary |
|---|---|---|---|---|
| Emily | Software Engineering | 2 | 4200 | 4200 |
| Boomily | Information Systems | 4.3 | 1000 | 1000000 |

**e. Method to remove rows containing missing data in subset of columns**

*Data.dropNull(new String[] {“Department”,”Actual graduation salary”})*

| Name | Department | CurrentCGPA | Expected Graduation | Actual graduation salary |
|---|---|---|---|---|
| Woof | | 2 | 4200 | |
| Meow | | 3.7 | 1000000 | 1000 |
| LWY | Information Systems | | 1000 | 1000000 |

| Name | Department | CurrentCGPA | Expected Graduation | Actual graduation salary |
|---|---|---|---|---|
| Meow | | 3.7 | 1000000 | 1000 |
| LWY | Information Systems | | 1000 | 1000000 |

3. Statistics package and imputers

   **a. Method compute variance, standard deviation, min, max,mean, median, mode and range of a column** (Non numeric columns will only have mode)

   **b. Method to fill in missing values of specified columns with a specified value.**

4. Scalers

   *Example values = {3.48, 2.30, 3.61, 3.16, 3.56, 2.9, 3.99, 4.87, 3.91, 6.28}*

   **a. Method to perform Standard Scaling.**

   Standard scaling is subtracting the mean from all values in the column and dividing by the standard deviation.

   *Output if done on example = {-0.31070398, -1.43533803, -0.18680362, -0.61568949, -0.23445761,*

   *-0.86349021, 0.17536667, 1.0140768 , 0.09912029, 2.35791918}*

   **b. Method to perform Min Max Scaling**

   Min max scaling is subtracting the min from all values in the column and dividing by the range

   *Output if done on example = {0.29648241, 0.000, 0.32914573, 0.2160804 , 0.31658291,*

   *0.15075377, 0.42462312, 0.64572864, 0.40452261, 1.000}*

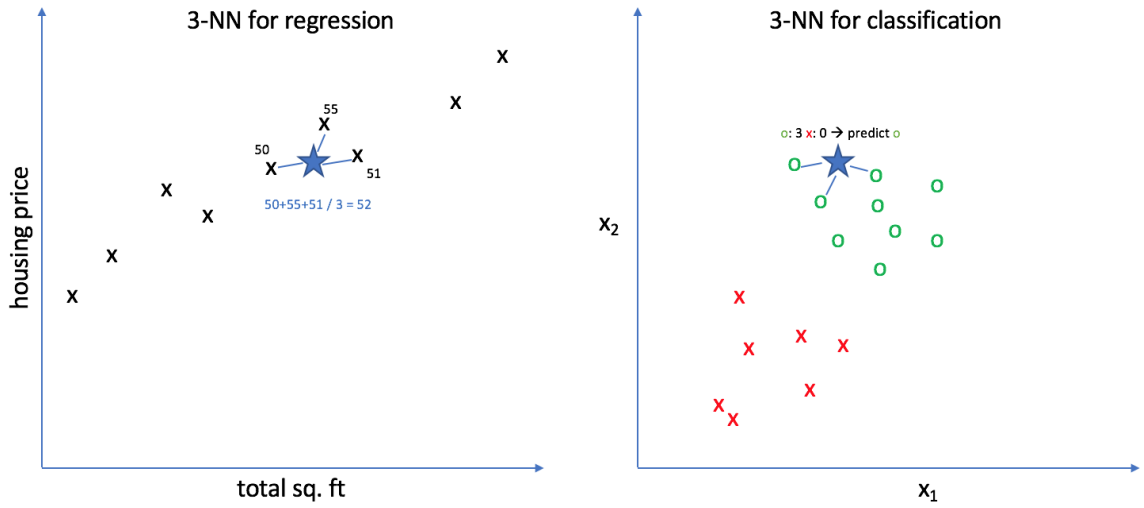
**The proceeding requirements can assume the DataFrame is made up entirely of float columns.**

5. K-Nearest Neighbors (k-nn)

   K-nearest neighbors is a simple prediction algorithm that uses k-nearest known instances to try and predict unknown instances. For this question, do k-nn based on euclidean distance. Euclidean distance between 2 points is the root mean squared of differences between 2 points.

   Regressor - output the mean of the instances

   Classification - output the mode of the instances




a. Method to impute values for a column using **k-nn regressor** using subset of other columns.

b. Method to impute values for a column using **k-nn classifier** using subset of other columns**.**

6. Error metric

   Being able to predict is no use if you don't know how often your predictions are correct. Implement at least **2 error metrics for the regressor** and **2 error metrics for the classifier**.

   Donald doesn't like errors so he isn't going to be particular about what you choose, he wants you to explore them on your own and implement what you think he should have.

Donald is also very particular about a few things:

1. Package management

   Donald is not a fan of one file spaghetti code. You are required to practice proper **subpackaging and file management** in your library. (At least split each numbered requirement into its own file. though it is recommended you group them into subpackages to more easily conceptualize your library, especially as you start implementing additional challenges)

2. Proper commenting and documentation

   Donald is a little dumb and is bad at logic. Do leave some **comments** so he doesn't get lost!

   (You don't need to put comments everywhere, just enough to describe what you intended with the method/subroutine.)

You should also document all the classes and methods in your library in one "**user manual**" to help Donald navigate and operate your library.

Use of **external libraries is prohibited** for implementations of the basic requirements of this question. They are allowed for implementations of the additional challenges as to not limit your creativity with what you can do, please use them **as a tool** in your implementation and not the implementation itself. External libraries here is defined as libraries outside of the java standard library.

**C. Additional Challenge**

1. Drag and drop with saving and loading pipelines

   To be frank, Donald isn't very good at this whole programming thing. If you had a drag and drop interface for constructing pipelines, Donald would get much more done quicker. Donald would also like to be able to save these pipelines and load them to continue where he left off.

2. Proof of concept for distributed/parallel computing

   Datasets in production these days can have millions of columns and billions of rows. For example, genomics data, high resolution image datasets and large arrays of IoT sensors. Analysis on these datasets usually require multiple computers churning through the dataset in parallel and sharing intermediate computations with each other. Donald would love to be able to use these datasets in his analysis as well. Maybe you could think of how you could convert the methods in the library you have created to use distributed computing.

3. Buzzword models

   k-nearest neighbors is definitely a powerful algorithm. However, Donalds friends think he's lame because he doesn't use any "advanced" models with fancy names. He wants to be cool. So cool. Can you help him out?

4. Different scalers

   Donald likes the scalers you've made so far. However, he's upset as the scalers seemed to make some of the datasets run worse! Looking into it, you realize that the scalers perform badly on skewed datasets as they amplify outliers. Are there other scalers out there to address this issue?

5. Read and write to database

Reading CSV files is nice. But a lot of data Donald is looking for is in databases waiting to be analyzed. Could you make the library support read and write from some of these databases? (SQL is enough. Feel free to explore non-SQL options too though.)

6. Web Scraping

   Donald is getting a little bored of analyzing the few CSV files he has, there are only so many available free interesting CSVs. But not all datasets come in nicely packaged CSV files. Many websites on the magical internet provide data for free but don't have nice APIs to query or charge a fee for API access. Could you help Donald out by creating a web scraper in your library to get him some more interesting CSVs? (Donald will be interested in whatever data you are interested in. As long as your eyes glow when talking about your web scraped dataset, Donald will pay you (with marks).)

7. Take off the training wheels!

   Constraints keep the project fun and allow you to focus on the fun parts. But Donald isn't exactly happy with your library limiting the datasets he can explore. Are you ready to face the quotes and commas nightmare? What about dates and timezones? Hashtables? yikes.

8. Anything else you find interesting

   There's a lot of other statistics transforms, encoders and visualisations you can do to help Donald on his Data Science journey. Feel free to bring your fresh innovative ideas to Donald too!

**D. Tips and comments**

1. If there is anything you are having trouble solving, try and think if sorting makes your problem simpler.
2. Code is read more than it is written. Don't be too smart. You may think it is cool to be able to code super fast with your 3 character variables or that you're uber cool because u wrote a one liner with multiple for loops and clauses, but your teammates and you from 1 week in the future will appreciate the extra readability of slowing down to name your methods and variables properly, practice proper spacing and write meaningful comments. Always code as if you are going to explain it one week later, because you are going to have to, even to yourself.
3. You might find GitHub useful for collaborating as it allows your team to sync code instead of sending files to each other on whatsapp. This assignment is pretty modular, so it should not be too hard to moderate the repo and will be a nice learning experience for using GitHub.

4. Methods can have similar names as long as they have different signatures. This may be useful for implementing methods for different types or with different numbers of parameters without confusing yourself by choosing a convoluted method name.
5. Dealing with floating point is something new programmers find very annoying, especially people coming from strong math backgrounds who firmly believe in real number precision. The fact that a binary computation system cannot deal with real numbers is one of the proofs that the computational model cannot solve every problem, as the computation model can only output integers, which is a much smaller set. However, as you become more attuned to computer science, you will start accepting this limitation and avoid fighting it when it's not crucial (for better or for worse). As a result, machine learning algorithms that deal with real numbers usually use float instead of double as the extra precision isn't worth the increased computation and memory. You may use double in your implementations for this assignment if you wish, but know that we won't penalize your precious marks over floating point errors.
6. You might find coding a matrix library with operation broadcasting helpful. Dealing with matrix operations is much easier to conceptualize than writing a pyramid of for loops all the time. This is one of the reasons why linear algebra has become so important in computer science; for loops are just hard for human intuition to deal with.
7. The constraint of only allowing columns with floats beyond requirement 4 is simply because you have not learnt about maps. In python they are called dictionaries which makes much more sense to the common person. Simply map non float columns to float with some sort of schema and you will be able to use the non float columns in your models.
8. The distributed computing additional feature requires good networking fundamentals and it is not recommended you try it unless you're prepared for that. For those who do, it is pretty simple in abstract. You basically need to have a master node that delegates tasks to its workers. The schema you choose to specify tasks and manage the worker-master relationship is up to your creativity. Do notify the question author if you want to attempt it as he would love for you to go through your idea with him. However, again, don't suffer over it please.
9. You can implement an unsupervised learning algorithm for the buzzword additional feature if you wish. However, be warned that unsupervised learning algorithms are not as straightforward as supervised ones as your goal is not always clear. But if you understand it quite well and think you can explain it, go ahead!
10. If you try a reinforcement learning algorithm, please notify the problem author. He would love to hear you explain it.
11. There's an amazing website called kaggle.com with a lot of data science competitions. If you found this assignment interesting. Feel free to continue learning about data science using kaggle.

**E. Questions**

If you are not clear about the project, you can contact ONG JACK MIN (ongjackm@gmail.com)

Subject: WIX1002 (Data Ducts)