Topic 5 : Data Alchemy - Transforming Raw Data into Predictive Gold

A. Introduction

Data pre-processing is a fundamental initial step in the field of machine learning, where raw and often messy data is transformed into a clean and structured format suitable for model training. While Python is the primary language of choice for data pre-processing due to its versatile libraries and tools, this assignment explores data pre-processing using Java—a language known for its robustness and versatility in various domains. This endeavor aims to broaden our horizons and uncover Java's potential in the crucial phase of data pre-processing, revealing both challenges and opportunities.

Machine learning plays a pivotal role in harnessing artificial intelligence, often relying on vast datasets. However, real-world data is typically imperfect, characterized by errors, missing values, outliers, and inconsistencies that can hinder subsequent analytical and modeling efforts. Data pre-processing steps act as unsung heroes in data science, identifying and rectifying these issues to ensure high-quality and trustworthy data for further analysis.

Python has rightfully earned its reputation as a data science powerhouse, known for its user-friendliness and strong community support. However, the real world often demands nuance. While Python excels in ease of use and community-driven solutions, Java stands out for its performance and memory management capabilities. It is at this intriguing intersection that we embark on a journey to uncover Java's untapped potential in data pre-processing. This exploration aims to illuminate how this stalwart language can complement and enhance data preparation efforts, ushering in new possibilities and ensuring utmost reliability in machine learning pursuits.

B. Problem statement

You have been recently hired by a data-driven company as a software developer specializing in data pre-processing. In your new role, your primary objective is to design and create a data pre-processing application that can efficiently clean, transform, and structure large datasets. The company relies heavily on data analytics, and your expertise in Java programming will be instrumental in streamlining data preparation processes, improving data quality, and ensuring that the data is ready for in-depth analysis and reporting. Your work will contribute to the company's ability to derive valuable insights from their data, drive data-driven decision-making, and maintain a competitive edge in their industry.

You are ***not allowed*** to use any pre-existing libraries or frameworks like Weka, Apache Spark, Apache Commons Math, Deeplearning4j, JSAT, Smile, MOA, or any similar tools. The assignment at hand requires you to create your own data pre-processing methods from scratch.

Click to access dataset.

Basic Requirement (8 marks)

1. Data Loading (0.5 marks)
    - Load the dataset and store them for further process.
2. Data Exploration (0.5 marks)
    - Explore the dataset to gain insights into its structure and characteristics.
    - Display the first few rows of the dataset to get a glimpse of its content.
    - Check for missing values in the dataset and decide how to handle them.
3. Data Cleaning (2 marks)
    - Handle missing values
        - Detect missing data points and decide on a strategy for imputing or removing them.
    - Remove duplicates
        - Iterate through the data and eliminate duplicate records if necessary.

- Feature Selection
  - Choosing the most relevant attributes from your dataset. Crucial when dealing with a large number of features to improve data analysis or machine learning model quality.

4. Data Transformation and Scaling (2 marks)
   - One-hot encoding
     - Convert categorical variables into binary columns.
   - Normalization
     - Scale feature values to a specific range, typically between 0 and 1 or -1 and 1.
   - Standardization
     - Transform feature values to have a mean of 0 and a standard deviation of 1.
5. Data Visualisation (1 mark)
   - Visualise your data in Command-Line Input (CLI) or can try out with Java's Swing or JavaFX libraries to design and build charts, graphs, and other visual representations of your data.
6. Data Splitting (1 mark)
   - Calculate the split ratio (e.g., 70% training, 30% testing).
   - Split data into training and testing set.
7. Save and Export Pre-processed Data (1 mark)
   - To save pre-processed data, use Java's I/O classes to create new files in the desired format (e.g., .csv) and write the pre-processed data to these files.

C. Sample input and output

1. Data Loading , Data Exploration and Data Visualisation
    - Note that there are missing values in ‘parking’ column.

```
String[][] data;
try {
    // load data from a csv file and store the data inside 2D array
    data = loadDataFromCSV( filename: "housepricing.csv");
} catch (IOException e) {
    throw new RuntimeException(e);
}
```

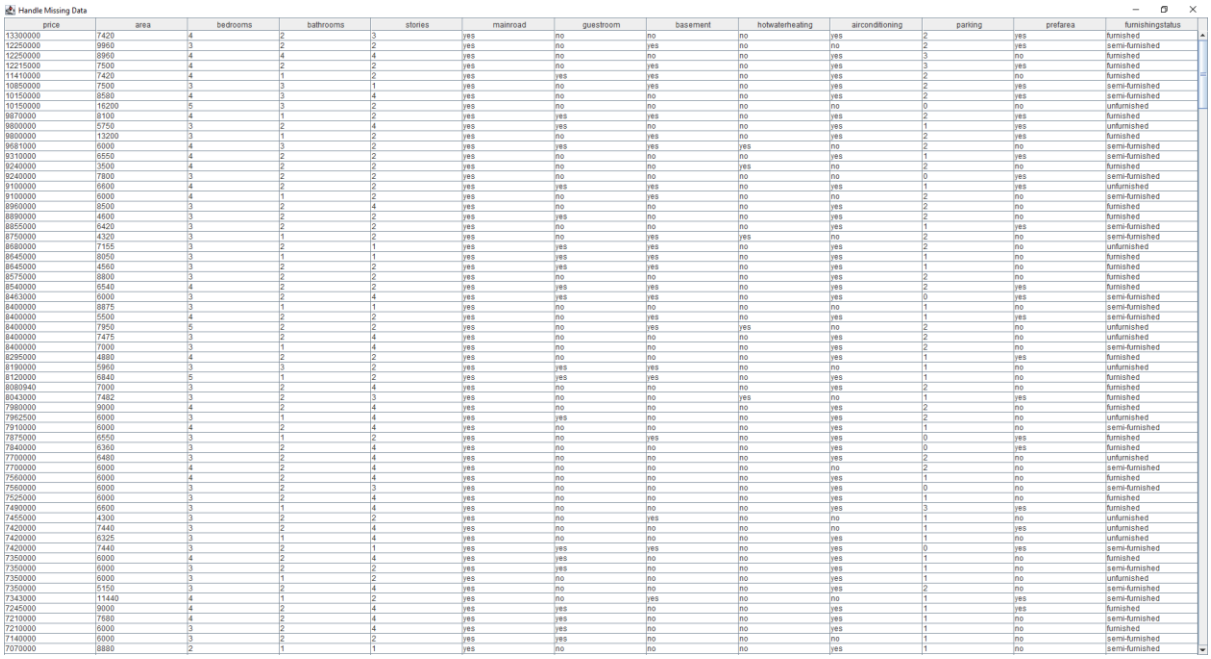
Data

price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished
10850000	7500	3	3	1	yes	no	yes	no	yes	2	yes	semi-furnished
10150000	8580	4	3	4	yes	no	no	no	yes	2	yes	semi-furnished
10150000	16200	5	3	2	yes	no	no	no	no		no	unfurnished
9870000	8100	4	1	2	yes	yes	yes	no	yes	2	yes	furnished
9800000	5750	3	2	4	yes	yes	no	no	yes	1	yes	unfurnished
9800000	13200	3	1	2	yes	no	yes	no	yes	2	yes	furnished
9681000	6000	4	3	2	yes	yes	yes	yes	no	2	no	semi-furnished
9310000	6550	4	2	2	yes	no	no	no	yes	1	yes	semi-furnished
9240000	3500	4	2	2	yes	no	no	yes	no	2	no	furnished
9240000	7800	3	2	2	yes	no	no	no	no		yes	semi-furnished
9100000	6600	4	2	2	yes	yes	yes	no	yes	1	yes	unfurnished
9100000	6000	4	1	2	yes	no	yes	no	no	2	no	semi-furnished
8960000	8500	3	2	4	yes	no	no	no	yes	2	no	furnished
8890000	4600	3	2	2	yes	yes	no	no	yes	2	no	furnished
8855000	6420	3	2	2	yes	no	no	no	yes	1	yes	semi-furnished
8750000	4320	3	1	2	yes	no	yes	yes	no	2	no	semi-furnished
8680000	7155	3	2	1	yes	yes	yes	no	yes	2	no	unfurnished
8645000	8050	3	1	1	yes	yes	yes	no	yes	1	no	furnished
8645000	4560	3	2	2	yes	yes	yes	no	yes	1	no	furnished
8575000	8800	3	2	2	yes	no	no	no	yes	2	no	furnished
8540000	6540	4	2	2	yes	yes	yes	no	yes	2	yes	furnished
8463000	6000	3	2	4	yes	yes	yes	no	yes		yes	semi-furnished
8400000	8875	3	1	1	yes	no	no	no	no	1	no	semi-furnished
8400000	5500	4	2	2	yes	no	yes	no	yes	1	yes	semi-furnished
8400000	7950	5	2	2	yes	no	yes	yes	no	2	no	unfurnished
8400000	7475	3	2	4	yes	no	no	no	yes	2	no	unfurnished
8400000	7000	3	1	4	yes	no	no	no	yes	2	no	semi-furnished
8295000	4880	4	2	2	yes	no	no	no	yes	1	yes	furnished
8190000	5960	3	3	2	yes	yes	yes	no	no	1	no	unfurnished
8120000	6840	5	1	2	yes	yes	yes	no	yes	1	no	furnished
8080940	7000	3	2	4	yes	no	no	no	yes	2	no	furnished
8043000	7482	3	2	3	yes	no	no	yes	no	1	yes	furnished
7980000	9000	4	2	4	yes	no	no	no	yes	2	no	furnished
7962500	6000	3	1	4	yes	yes	no	no	yes	2	no	unfurnished
7910000	6000	4	2	4	yes	no	no	no	yes	1	no	semi-furnished
7875000	6550	3	1	2	yes	no	yes	no	yes		yes	furnished
7840000	6360	3	2	4	yes	no	no	no	yes		yes	furnished
7700000	6480	3	2	4	yes	no	no	no	yes	2	no	unfurnished
7700000	6000	4	2	4	yes	no	no	no	no	2	no	semi-furnished
7560000	6000	4	2	4	yes	no	no	no	yes	1	no	furnished
7560000	6000	3	2	3	yes	no	no	no	yes		no	semi-furnished
7525000	6000	3	2	4	yes	no	no	no	yes	1	no	furnished
7490000	6600	3	1	4	yes	no	no	no	yes	3	yes	furnished
7455000	4300	3	2	2	yes	no	yes	no	no	1	no	unfurnished
7420000	7440	3	2	4	yes	no	no	no	no	1	yes	unfurnished
7420000	6325	3	1	4	yes	no	no	no	yes	1	no	unfurnished
7420000	7440	3	2	1	yes	yes	yes	no	yes		yes	semi-furnished
7350000	6000	4	2	4	yes	yes	no	no	yes	1	no	furnished
7350000	6000	3	2	2	yes	yes	no	no	yes	1	no	semi-furnished
7350000	6000	3	1	2	yes	no	no	no	yes	1	no	unfurnished
7350000	5150	3	2	4	yes	no	no	no	yes	2	no	semi-furnished
7343000	11440	4	1	2	yes	no	yes	no	no	1	yes	semi-furnished
7245000	9000	4	2	4	yes	yes	no	no	yes	1	yes	furnished
7210000	7680	4	2	4	yes	yes	no	no	yes	1	no	semi-furnished
7210000	6000	3	2	4	yes	yes	no	no	yes	1	no	furnished
7140000	6000	3	2	2	yes	yes	no	no	no	1	no	semi-furnished
7070000	8880	2	1	1	yes	no	no	no	yes	1	no	semi-furnished

2. Data Cleaning – handle missing value

- Note that the missing values in ‘parking’ feature is replaced by 0.

```
String[][] handledData;
handledData = handleMissingValue(data);
```



Handle Missing Data

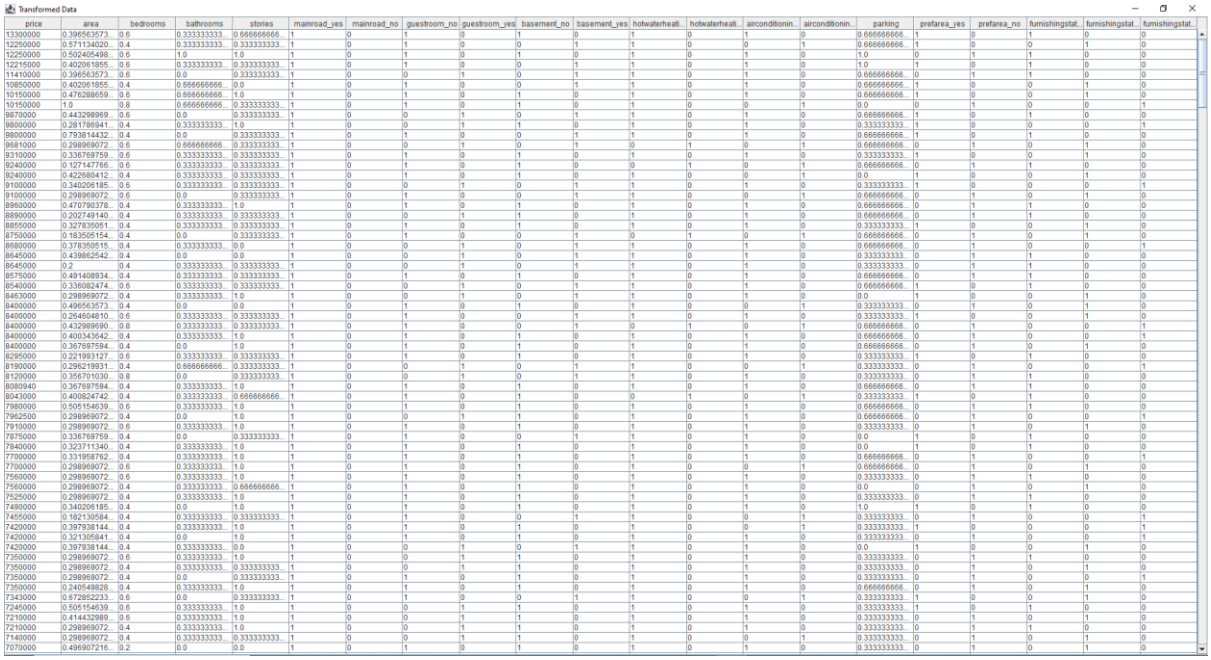
price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished
10850000	7500	3	3	1	yes	no	yes	no	yes	2	yes	semi-furnished
10150000	8580	4	3	4	yes	no	no	no	yes	2	yes	semi-furnished
10150000	16200	5	3	2	yes	no	no	no	no	0	no	unfurnished
9870000	8100	4	1	2	yes	yes	yes	no	yes	2	yes	furnished
9800000	5750	3	2	4	yes	yes	no	no	yes	1	yes	unfurnished
9800000	13200	3	1	2	yes	no	yes	no	yes	2	yes	furnished
9681000	6000	4	3	2	yes	yes	yes	yes	no	2	no	semi-furnished
9310000	6550	4	2	2	yes	no	no	no	yes	1	yes	semi-furnished
9240000	3500	4	2	2	yes	no	no	yes	no	2	no	furnished
9240000	7800	3	2	2	yes	no	no	no	no	0	yes	semi-furnished
9100000	6600	4	2	2	yes	yes	yes	no	yes	1	yes	unfurnished
9100000	6000	4	1	2	yes	no	yes	no	no	2	no	semi-furnished
8960000	8500	3	2	4	yes	no	no	no	yes	2	no	furnished
8890000	4600	3	2	2	yes	yes	no	no	yes	2	no	furnished
8855000	6420	3	2	2	yes	no	no	no	yes	1	yes	semi-furnished
8750000	4320	3	1	2	yes	no	yes	yes	no	2	no	semi-furnished
8680000	7155	3	2	1	yes	yes	yes	no	yes	2	no	unfurnished
8645000	8050	3	1	1	yes	yes	yes	no	yes	1	no	furnished
8645000	4560	3	2	2	yes	yes	yes	no	yes	1	no	furnished
8575000	8800	3	2	2	yes	no	no	no	yes	2	no	furnished
8540000	6540	4	2	2	yes	yes	yes	no	yes	2	yes	furnished
8463000	6000	3	2	4	yes	yes	yes	no	yes	0	yes	semi-furnished
8400000	8875	3	1	1	yes	no	no	no	no	1	no	semi-furnished
8400000	5500	4	2	2	yes	no	yes	no	yes	1	yes	semi-furnished
8400000	7950	5	2	2	yes	no	yes	yes	no	2	no	unfurnished
8400000	7475	3	2	4	yes	no	no	no	yes	2	no	unfurnished
8400000	7000	3	1	4	yes	no	no	no	yes	2	no	semi-furnished
8295000	4880	4	2	2	yes	no	no	no	yes	1	yes	furnished
8190000	5960	3	3	2	yes	yes	yes	no	no	1	no	unfurnished
8120000	6840	5	1	2	yes	yes	yes	no	yes	1	no	furnished
8080940	7000	3	2	4	yes	no	no	no	yes	2	no	furnished
8043000	7482	3	2	3	yes	no	no	yes	no	1	yes	furnished
7980000	9000	4	2	4	yes	no	no	no	yes	2	no	furnished
7962500	6000	3	1	4	yes	yes	no	no	yes	2	no	unfurnished
7910000	6000	4	2	4	yes	no	no	no	yes	1	no	semi-furnished
7875000	6550	3	1	2	yes	no	yes	no	yes	0	yes	furnished
7840000	6360	3	2	4	yes	no	no	no	yes	0	yes	furnished
7700000	6480	3	2	4	yes	no	no	no	yes	2	no	unfurnished
7700000	6000	4	2	4	yes	no	no	no	no	2	no	semi-furnished
7560000	6000	4	2	4	yes	no	no	no	yes	1	no	furnished
7560000	6000	3	2	3	yes	no	no	no	yes	0	no	semi-furnished
7525000	6000	3	2	4	yes	no	no	no	yes	1	no	furnished
7490000	6600	3	1	4	yes	no	no	no	yes	3	yes	furnished
7455000	4300	3	2	2	yes	no	yes	no	no	1	no	unfurnished
7420000	7440	3	2	4	yes	no	no	no	no	1	yes	unfurnished
7420000	6325	3	1	4	yes	no	no	no	yes	1	no	unfurnished
7420000	7440	3	2	1	yes	yes	yes	no	yes	0	yes	semi-furnished
7350000	6000	4	2	4	yes	yes	no	no	yes	1	no	furnished
7350000	6000	3	2	2	yes	yes	no	no	yes	1	no	semi-furnished
7350000	6000	3	1	2	yes	no	no	no	yes	1	no	unfurnished
7350000	5150	3	2	4	yes	no	no	no	yes	2	no	semi-furnished
7343000	11440	4	1	2	yes	no	yes	no	no	1	yes	semi-furnished
7245000	9000	4	2	4	yes	yes	no	no	yes	1	yes	furnished
7210000	7680	4	2	4	yes	yes	no	no	yes	1	no	semi-furnished
7210000	6000	3	2	4	yes	yes	no	no	yes	1	no	furnished
7140000	6000	3	2	2	yes	yes	no	no	no	1	no	semi-furnished
7070000	8880	2	1	1	yes	no	no	no	yes	1	no	semi-furnished

3. Data Transformation and Scaling

- The numerical value is normalized and the categorical value is encoded.

```
String[][] normalizedData;
//define which feature contain numerical data
String[] numericalData = {"area", "bedrooms", "bathrooms", "stories", "parking"};
normalizedData = normalizeNumericFeatures(handledData, numericalData);

String[][] transformedData;
//define which feature contain categorical data
String[] categoricalData = {"mainroad", "guestroom", "basement", "hotwaterheating", "airconditioning", "prefarea", "furnishingstatus"};
transformedData = oneHotEncode(normalizedData, categoricalData);
```



Transformed Data

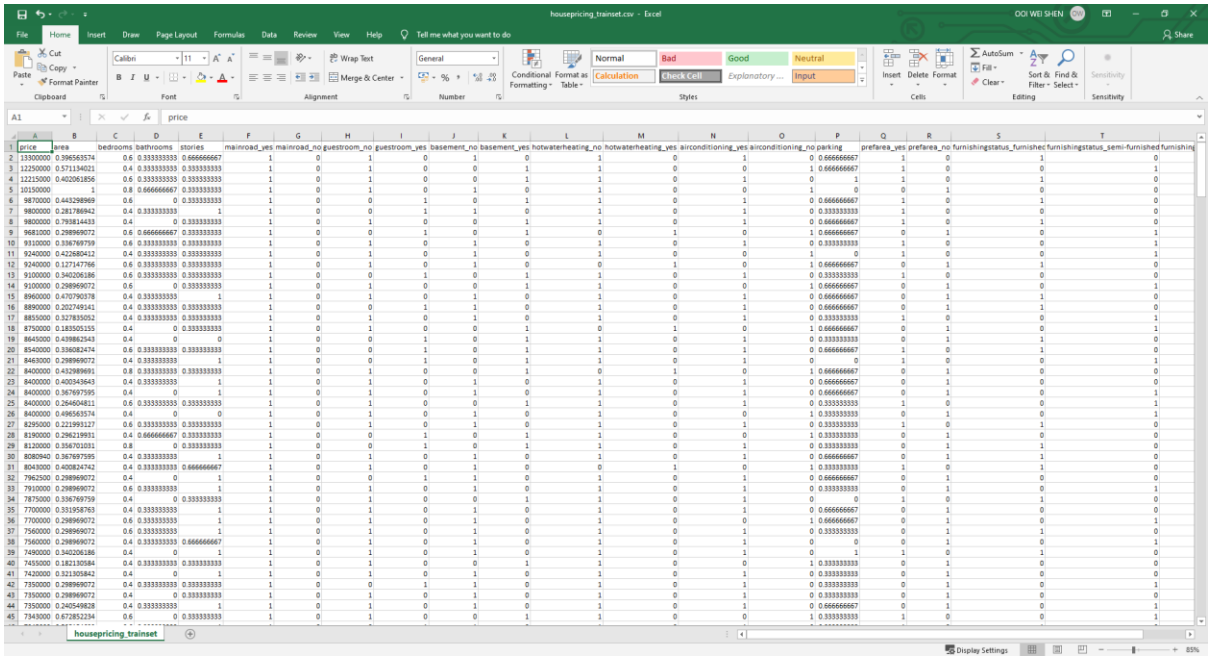
price	area	bedrooms	bathrooms	stories	mainroad_yes	mainroad_no	guestroom_no	guestroom_yes	basement_no	basement_yes	hotwaterheati...	hotwaterheati...	airconditionin...	airconditionin...	parking	prefarea_yes	prefarea_no	furnishingstat...	furnishingstat...	furnishingstat...
13300000	0.396563573...	0.6	0.333333333...	0.666666666...	1	0	1	0	1	0	1	0	1	0	0.666666666...	1	0	1	0	0
12250000	0.571134020...	0.4	0.333333333...	0.333333333...	1	0	1	0	0	1	1	0	0	1	0.666666666...	1	0	0	1	0
12250000	0.502405498...	0.6	1.0	1.0	1	0	1	0	1	0	1	0	1	0	1.0	1	0	1	0	0
12215000	0.402061855...	0.6	0.333333333...	0.333333333...	1	0	1	0	0	1	1	0	1	0	1.0	1	0	1	0	0
11410000	0.396563573...	0.6	0.0	0.333333333...	1	0	0	1	0	1	1	0	1	0	0.666666666...	0	1	1	0	0
10850000	0.402061855...	0.4	0.666666666...	0.0	1	0	1	0	0	1	1	0	1	0	0.666666666...	1	0	0	1	0
10150000	0.476288659...	0.6	0.666666666...	1.0	1	0	1	0	1	0	1	0	1	0	0.666666666...	1	0	0	1	0
10150000	1.0	0.8	0.666666666...	0.333333333...	1	0	1	0	1	0	1	0	0	1	0.0	0	1	0	0	1
9870000	0.443298969...	0.6	0.0	0.333333333...	1	0	0	1	0	1	1	0	1	0	0.666666666...	1	0	1	0	0
9800000	0.281786941...	0.4	0.333333333...	1.0	1	0	0	1	1	0	1	0	1	0	0.333333333...	1	0	0	0	1
9800000	0.793814432...	0.4	0.0	0.333333333...	1	0	1	0	0	1	1	0	1	0	0.666666666...	1	0	1	0	0
9681000	0.298969072...	0.6	0.666666666...	0.333333333...	1	0	0	1	0	1	0	1	0	1	0.666666666...	0	1	0	1	0
9310000	0.336769759...	0.6	0.333333333...	0.333333333...	1	0	1	0	1	0	1	0	1	0	0.333333333...	1	0	0	1	0
9240000	0.127147766...	0.6	0.333333333...	0.333333333...	1	0	1	0	1	0	0	1	0	1	0.666666666...	0	1	1	0	0
9240000	0.422680412...	0.4	0.333333333...	0.333333333...	1	0	1	0	1	0	1	0	0	1	0.0	1	0	0	1	0
9100000	0.340206185...	0.6	0.333333333...	0.333333333...	1	0	0	1	0	1	1	0	1	0	0.333333333...	1	0	0	0	1
9100000	0.298969072...	0.6	0.0	0.333333333...	1	0	1	0	0	1	1	0	0	1	0.666666666...	0	1	1	0	0
8960000	0.470790378...	0.4	0.333333333...	1.0	1	0	1	0	1	0	1	0	1	0	0.666666666...	0	1	1	0	0
8890000	0.202749140...	0.4	0.333333333...	0.333333333...	1	0	0	1	1	0	1	0	1	0	0.666666666...	0	1	1	0	0
8855000	0.327835051...	0.4	0.333333333...	0.333333333...	1	0	1	0	1	0	1	0	1	0	0.333333333...	1	0	0	1	0
8750000	0.183505154...	0.4	0.0	0.333333333...	1	0	1	0	0	1	0	1	0	1	0.666666666...	0	1	0	1	0
8680000	0.378350515...	0.4	0.333333333...	0.0	1	0	0	1	0	1	1	0	1	0	0.666666666...	0	1	0	0	1
8645000	0.439862542...	0.4	0.0	0.0	1	0	1	0	1	0	1	0	1	0	0.333333333...	0	1	1	0	0
8645000	0.2	0.4	0.333333333...	0.333333333...	1	0	0	1	1	0	1	0	1	0	0.333333333...	0	1	1	0	0
8575000	0.491408934...	0.4	0.333333333...	0.333333333...	1	0	1	0	1	0	1	0	1	0	0.666666666...	0	1	1	0	0
8540000	0.336082474...	0.6	0.333333333...	0.333333333...	1	0	0	1	0	1	1	0	1	0	0.666666666...	1	0	1	0	0
8463000	0.298969072...	0.4	0.333333333...	1.0	1	0	0	1	0	1	1	0	0	1	0.0	1	0	0	1	0
8400000	0.496563573...	0.4	0.0	0.0	1	0	1	0	1	0	1	0	0	1	0.333333333...	0	1	0	1	0
8400000	0.264604810...	0.6	0.333333333...	0.333333333...	1	0	1	0	0	1	1	0	1	0	0.333333333...	1	0	0	1	0
8400000	0.432989690...	0.8	0.333333333...	0.333333333...	1	0	1	0	0	1	0	1	0	1	0.666666666...	0	1	0	0	1
8400000	0.400343642...	0.4	0.333333333...	1.0	1	0	1	0	1	0	1	0	1	0	0.666666666...	0	1	0	0	1
8400000	0.367697594...	0.4	0.0	1.0	1	0	1	0	1	0	1	0	1	0	0.666666666...	0	1	0	1	0
8295000	0.221993127...	0.6	0.333333333...	0.333333333...	1	0	1	0	1	0	1	0	1	0	0.333333333...	1	0	1	0	0
8190000	0.296219931...	0.4	0.666666666...	0.333333333...	1	0	0	1	0	1	1	0	0	1	0.333333333...	0	1	0	0	1
8120000	0.356701030...	0.8	0.0	0.333333333...	1	0	0	1	0	1	1	0	1	0	0.333333333...	0	1	1	0	0
8080940	0.367697594...	0.4	0.333333333...	1.0	1	0	1	0	1	0	1	0	1	0	0.666666666...	0	1	1	0	0
8043000	0.400824742...	0.4	0.333333333...	0.666666666...	1	0	1	0	1	0	0	1	0	1	0.333333333...	1	0	1	0	0
7980000	0.505154639...	0.6	0.333333333...	1.0	1	0	1	0	1	0	1	0	1	0	0.666666666...	0	1	1	0	0
7962500	0.298969072...	0.4	0.0	1.0	1	0	0	1	1	0	1	0	1	0	0.666666666...	0	1	0	0	1
7910000	0.298969072...	0.6	0.333333333...	1.0	1	0	1	0	1	0	1	0	1	0	0.333333333...	0	1	0	1	0
7875000	0.336769759...	0.4	0.0	0.333333333...	1	0	1	0	0	1	1	0	1	0	0.0	1	0	1	0	0
7840000	0.323711340...	0.4	0.333333333...	1.0	1	0	1	0	1	0	1	0	1	0	0.0	1	0	1	0	0
7700000	0.331958762...	0.4	0.333333333...	1.0	1	0	1	0	1	0	1	0	1	0	0.666666666...	0	1	0	0	1
7700000	0.298969072...	0.6	0.333333333...	1.0	1	0	1	0	1	0	1	0	0	1	0.666666666...	0	1	0	1	0
7560000	0.298969072...	0.6	0.333333333...	1.0	1	0	1	0	1	0	1	0	1	0	0.333333333...	0	1	1	0	0
7560000	0.298969072...	0.4	0.333333333...	0.666666666...	1	0	1	0	1	0	1	0	1	0	0.0	0	1	0	1	0
7525000	0.298969072...	0.4	0.333333333...	1.0	1	0	1	0	1	0	1	0	1	0	0.333333333...	0	1	1	0	0
7490000	0.340206185...	0.4	0.0	1.0	1	0	1	0	1	0	1	0	1	0	1.0	1	0	1	0	0
7455000	0.182130584...	0.4	0.333333333...	0.333333333...	1	0	1	0	0	1	1	0	0	1	0.333333333...	0	1	0	0	1
7420000	0.397938144...	0.4	0.333333333...	1.0	1	0	1	0	1	0	1	0	0	1	0.333333333...	1	0	0	0	1
7420000	0.321305841...	0.4	0.0	1.0	1	0	1	0	1	0	1	0	1	0	0.333333333...	0	1	0	0	1
7420000	0.397938144...	0.4	0.333333333...	0.0	1	0	0	1	0	1	1	0	1	0	0.0	1	0	0	1	0
7350000	0.298969072...	0.6	0.333333333...	1.0	1	0	0	1	1	0	1	0	1	0	0.333333333...	0	1	1	0	0
7350000	0.298969072...	0.4	0.333333333...	0.333333333...	1	0	0	1	1	0	1	0	1	0	0.333333333...	0	1	0	1	0
7350000	0.298969072...	0.4	0.0	0.333333333...	1	0	1	0	1	0	1	0	1	0	0.333333333...	0	1	0	0	1
7350000	0.240549828...	0.4	0.333333333...	1.0	1	0	1	0	1	0	1	0	1	0	0.666666666...	0	1	0	1	0
7343000	0.572852233...	0.6	0.0	0.333333333...	1	0	1	0	0	1	1	0	0	1	0.333333333...	1	0	0	1	0
7245000	0.505154639...	0.6	0.333333333...	1.0	1	0	0	1	1	0	1	0	1	0	0.333333333...	1	0	1	0	0
7210000	0.414432989...	0.6	0.333333333...	1.0	1	0	0	1	1	0	1	0	1	0	0.333333333...	0	1	0	1	0
7210000	0.298969072...	0.4	0.333333333...	1.0	1	0	0	1	1	0	1	0	1	0	0.333333333...	0	1	1	0	0
7140000	0.298969072...	0.4	0.333333333...	0.333333333...	1	0	0	1	1	0	1	0	0	1	0.333333333...	0	1	0	1	0
7070000	0.496907216...	0.2	0.0	0.0	1	0	1	0	1	0	1	0	1	0	0.333333333...	0	1	0	1	0

4. Data Splitting and Export Pre-processed Data

    - The data has been divided into a training set and a test set, and it has been saved for further processing.

```
String[][][] train_test_split = splitData(transformedData, trainRatio: 0.7);

try {
    //export training set in csv form
    exportToCsv(train_test_split[0], csvFileName: "housepricing_trainset.csv");
} catch (IOException e) {
    e.printStackTrace();
}
```



D. Additional challenge (4 marks)

1. Data Inclusiveness
    - Real-world datasets are commonly accessible in diverse file formats, such as CSV and DOCX. To ensure thorough data pre-processing, it is essential to have the capability to seamlessly process all types of datasets, without requiring additional custom logic.
    - Here are a few websites that you can access to datasets:
        - Kaggle (https://www.kaggle.com/datasets)
        - GitHub datasets (https://github.com/datasets)
        - UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/index.php)
        - Data.gov (https://www.data.gov/)
2. Graphical User Interface (GUI)
    - Develop a clear and user-friendly Graphical User Interface (GUI) to enhance the user experience and ease the configuration and monitoring of data pre-processing tasks.
3. Create Your Own Library
    - Explore the feasibility of creating your own Java library to facilitate easy and efficient data pre-processing for future use, leveraging Java's object-oriented capabilities.
4. Explore More Feature of Data Pre-processing
    - Explore features like Merging Datasets, Feature Engineering, Outlier Detection and Handling, Data Validation and other relevant features yourself.
5. Sorting and searching function
    - Make your user able to sort and search the data according to their preferences.
6. Add other optional features that come to your mind to make your assignment stand out from the rest to impress your demonstrator and lecturer. Don't hesitate to explore creative solutions and new functionalities.

E. Comments

1. Use Version Control with Git and GitHub
    - Consider using Git and GitHub for version control throughout the development of your Java program. This will make it easier to track changes, collaborate with team members, and ensure that your code is backed up.
2. Explore Data Structures like ArrayList for Improved Data Management
    - In this assignment, while using a standard array is acceptable and functional for storing your data, I strongly recommend exploring data structures like ArrayList to enhance your program's data management capabilities.
3. Encourage Encapsulation and Object-Oriented Principles
    - While implementing the program, consider incorporating object-oriented programming (OOP) principles and encapsulation.
4. Mastering your File I/O and Flow Control Concepts, especially Repetition through Loops
    - Strong flow control, achieved through effective repetition with loops, is the backbone of efficient data pre-processing. Embrace the power of iteration to maintain data integrity and streamline the preparation of your dataset for machine learning.
5. Exception Handling
    - Utilizes 'try,' 'catch,' and 'throw' blocks to gracefully manage exceptions, thereby averting program crashes and enabling the implementation of custom error-handling logic for immediate error identification.
6. Data pre-processing is mandatory for the course Machine Learning which you will take next sem. Why not have a look first before the course start while working on the assignment. I think it is useful as a preparation for the course next sem.
7. You can explore additional suitable Datasets independently. The example Datasets provided are for my example purposes only.

F. Contact Me

If you have any questions about the assignment you may contact me (Ooi Wei Shen) via

- WhatsApp (012-329 0365) or
- Email at weishenooi2003@gmail.com

I will try my best to assist you all in the assignment. ***Good Luck Have Fun!***