Data Analyst Assessment: Employee Data Cleaning and Analysis

Objective: This assessment tests your ability to clean, transform, and analyze a messy dataset using Python and Pandas. You will be working with a dataset containing employee information. Your goal is to apply your data manipulation skills to address data quality issues and extract meaningful insights.

Dataset: `empd.csv` (Attached with the email)

Instructions:

1. For each question below, provide the Python/Pandas code you used to solve the problem.
2. Ensure your code is well-commented and easy to understand.
3. Where applicable, provide a brief explanation of your approach and any assumptions you made.
4. Submit your code in a single Python script or Jupyter Notebook (`.py` or `.ipynb` file).
5. **Time Limit:** 1 Day (Submission Closes At 14th Feb 11:59 PM)

Questions:

1. **Missing Salary Imputation:**
    - Write Python/Pandas code to impute missing `salary` values using the *median* salary for each `department` .
2. **Duplicate Employee ID Check:**
    - Write Python/Pandas code to identify and list all `employee_id` values that appear more than once in the DataFrame. Print the duplicate `employee_id` values to the console.
3. **Performance Score Conversion:**
    - Write Python/Pandas code to convert the `performance_score` column to an integer type. Replace `None` values with `-1` . Handle any potential errors during the conversion (e.g., non-numeric strings) by replacing them with `-1` as well.
4. **Date Format Standardization:**
    - Write Python/Pandas code to convert the `join_date` column to a consistent `YYYY-MM-DD` format and create a new `join_year` column (as an integer). Handle any errors that may occur during the date conversion by skipping the row and printing an error message (including the problematic `join_date` value).

5. **Department Typo Correction:**
    - Write Python/Pandas code to correct the typos in the `department` column.
6. **Salary Outlier Removal:**
    - Write Python/Pandas code to remove rows where the `salary` is more than 3 standard deviations from the mean salary. Calculate the mean and standard deviation *before* removing any outliers.
7. **Email Domain Extraction:**
    - Write Python/Pandas code to extract the domain from each email address and store it in a new column called `email_domain`. Handle any cases where the email address is invalid or missing (e.g., by assigning `None` to the `email_domain` column in those rows).
8. **Salary Correlation Calculation:**
    - Write Python/Pandas code to calculate the Pearson correlation coefficient between `salary` and `join_year` (after converting `join_year` to a numerical type). Print the correlation coefficient to the console. Handle cases where the correlation cannot be calculated due to insufficient data.
9. **Average Salary Aggregation:**
    - Write Python/Pandas code to calculate the average salary per `department` and `job_level`, and display the results in a pivot table. Print the pivot table to the console.
10. **Duplicate Row Removal:**
    - Write Python/Pandas code to remove all duplicate rows from the DataFrame, considering all columns. After removing the duplicates, write code to verify that the removal was successful by checking if there are any remaining duplicate rows. Print a message indicating whether or not duplicates were successfully removed.

Evaluation Criteria:

- **Correctness:** Accuracy of the code in addressing the problem.
- **Code Quality:** Readability, comments, and adherence to coding best practices.
- **Error Handling:** Ability to handle potential errors and edge cases.
- **Efficiency:** Efficiency of the code in terms of performance.
- **Clarity:** Clarity of explanations and justifications provided.

Submission:

Submit a single `.py` or `.ipynb` file containing your solutions to all the questions. Include your name and contact information in the file header.