**Introduction to Data Science** **Assignment 3**

Instructors: **Dr. Bahrak, Dr. Yaghoobzadeh**

TAs: Ali Karimi, Hesam Ramezanian, Mobina Mehrazar

Deadline: Ordibehesht 19th, 11:59PM

## Introduction

In this assignment, you will work on three real-world machine learning challenges, each requiring a different modeling approach: classification, regression, and recommendation. Each task has its own dataset, and you will participate in a dedicated Kaggle competition linked to that task. Your objective is to build and submit machine learning models that produce high-quality predictions based on your data understanding and processing skills.

For Task 1, you will develop a binary classification model to predict the survival status of cancer patients.
For Task 2, you will build a regression model to predict the combined number of casual and registered bike rentals on a given day based on provided features.
For Task 3, you will design a movie recommendation system and predict user ratings.

Each task emphasizes real-world data, where understanding and preparing the data is just as important as model selection. Your submissions will be evaluated based on performance metrics defined on the corresponding Kaggle competition pages.

Note: You are strictly prohibited from using any deep learning models or pre-trained models (whether fine-tuned or not). Only traditional machine learning algorithms are allowed. However, you are free to use any Python library that supports classic ML workflows (e.g., scikit-learn, XGBoost, LightGBM, etc.).

## Assignment Details and Instructions

Each of the three tasks in this assignment has a corresponding Kaggle competition, where the datasets required for that task are hosted. You can participate by clicking on the Kaggle competition link provided at the end of each task description.

**What You Must Do:**

- Click the competition link at the bottom of each task to join the related Kaggle competition.
  Read the task instructions, submission rules, and evaluation metrics carefully on the Kaggle competition page.
- Once your model is ready, generate predictions on the test set and save them in a CSV file. The expected submission format is described on the competition page (usually a CSV file with columns like `id` and `label`).
- Upload the prediction file to the Kaggle competition page for each task.
- Your model's public score will be calculated based on 30% of the test data. After the competition deadline, your final rank will be determined using all of the test data (100%).
- The test labels are hidden — this is to prevent you from tuning your model on the test set. Avoid overfitting to the public leaderboard.
- Do NOT use deep learning models or architectures.
- Do NOT use or fine-tune pre-trained models.
- You are allowed to use traditional machine learning libraries, including but not limited to: scikit-learn, CatBoost, XGBoost, and LightGBM.
- The most critical factor in improving your model's accuracy is good data preprocessing and feature engineering. We strongly recommend investing time in understanding and transforming your data.

# Task 1 : Classification

## Objective:

This project challenges you to develop a robust machine learning model to predict survival outcomes for cancer patients. Using a dataset containing patient demographics, diagnosis details, treatment history, and various examination results, your primary goal is to accurately classify the label(Survival_Status) for each individual, where 1 signifies 'Alive' and 0 signifies 'Deceased'.

## Process Overview:

Success in this project hinges on a methodical approach encompassing data exploration, preparation, modeling, and evaluation.

1. **Data Exploration & Understanding:** Begin by thoroughly analyzing the dataset. Investigate the relationships between features (like Birth_Date, Weight, Cancer_Type, Stage_at_Diagnosis, Treatment History, etc.) and the Survival_Status target variable to gain insights into factors influencing patient outcomes.
2. **Preprocessing & Feature Engineering:** Effective data preparation is crucial for building a high-performing model. Apply appropriate techniques such as handling missing values, encoding categorical features (Urban_Rural, Occupation, etc.), scaling numerical data, managing date information, and potentially engineering new features to better capture predictive signals.
3. **Model Development & Selection:** Experiment with various binary classification algorithms suitable for this task (e.g., Logistic Regression, SVM, Decision Trees). Compare their performance using appropriate validation strategies and select the model that yields the highest **Accuracy** on your validation set.
4. **Comprehensive Evaluation:** While **Accuracy** is the primary metric for final scoring, you must also calculate and log **Precision, Recall, and F1-Score** during your model development process. These metrics provide a more nuanced understanding of your model's strengths and weaknesses.

## Prediction & Submission:

Your final, trained model will be used to predict the Survival_Status for a separate test dataset where the true labels are withheld.

- **Output:** Submit your predictions as a **CSV file**.
- **Format:** The CSV must contain exactly two columns:
  - id: The unique identifier for each test instance.

- ○ label: Your model's predicted value (0 or 1).

## Dataset Overview:

| Feature Name | Description |
|---|---|
| **Birth_Date** | The patient's date of birth. |
| **Weight** | The patient's weight measurement. |
| **Height** | The patient's height. |
| **Urban_Rural** | Indicates whether the patient lives in an urban or rural area. |
| **Occupation** | The patient’s profession. |
| **Insurance_Type** | The type of health insurance the patient holds. |
| **Family_History** | Indicates whether there is a family history of cancer. |
| **Cancer_Type** | Specifies the type of cancer diagnosed in the patient. |
| **Stage_at_Diagnosis** | Describes the cancer stage at the time of diagnosis. |
| **Diagnosis_Date** | The date when the cancer diagnosis was made. |
| **Symptoms** | The reported symptoms at diagnosis or during disease progression. |
| **Tumor_Size** | The size of the tumor. |
| **Surgery_Date** | The date on which surgery was performed. |
| **Chemotherapy_Drugs** | Lists the chemotherapy drugs administered to the patient. |
| **Radiation_Sessions** | The count of radiation therapy sessions the patient received. |

| | |
|---|---|
| **Immunotherapy** | Indicates whether the patient underwent immunotherapy treatment. |
| **Targeted_Therapy** | Indicates whether the patient received targeted therapy, designed to attack specific cancer cells. |
| **Recurrence_Status** | Indicates whether the cancer has recurred following initial treatment. |
| **Smoking_History** | Information about the patient's smoking habits. |
| **Alcohol_Use** | Details regarding the patient's alcohol consumption. |
| **label(Survival_Status)** | The current survival outcome of the patient (e.g., 1(alive) or 0(deceased)). |

To participate in the competition related to this task Click on the competition link: Kaggle Competition.

Good luck with the competition!

## Task 2 : Regression

For this assignment task, you'll work with a dataset that records daily bike rental activity, along with weather conditions, seasonal information, and calendar details. Your goal is to develop a regression model that predicts the `total_users`.

While the data is collected daily, this is not a time series forecasting task. Instead, you'll use time-related features (such as season, weekday, and holiday indicators) as inputs to help explain patterns in bike rental behavior.

Begin by exploring the dataset to identify how features such as season, temperature, humidity, and day type affect rental activity. Apply appropriate preprocessing techniques, including encoding categorical variables and scaling numerical features.

Since the number of features in this task is relatively high, selecting the most relevant features that contribute meaningfully to the regression performance is important. It is recommended to apply statistical methods for selecting high-quality features to improve the model's accuracy and robustness.

Next, experiment with a variety of traditional regression algorithms and select the one that performs best on the training data. Finally, train your model and use it to predict the number of users in the test dataset.

**Note:** The `total_users` column is the target variable and is missing in the test set. Your model should generate accurate numeric predictions for each day and output them in a CSV file with two columns: `id` and `label`.

| Column Name | Description |
|---|---|
| **id** | Unique identifier for each record (starting from 1) |
| **date** | The specific calendar date (DD-MM-YYYY) of the record |
| **season_id** | Categorical identifier for the quarter of the year based on the Gregorian calendar |
| **year** | Year of the record (e.g., 0 for 2018, 1 for 2019) |
| **month** | Numeric month of the year (1–12) |
| **is_holiday** | Indicates if the day is a holiday (1 = Yes, 0 = No) |
| **weekday** | Day of the week (0 = Tuesday, 6 = Monday) |
| **is_workingday** | Indicates if the day is a working day (1 = Yes, 0 = No) |

| | |
|---|---|
| **weather_condition** | Categorical weather rating |
| **temperature** | temperature in Celsius (continuous variable) |
| **feels_like_temp** | apparent temperature (what it feels like) |
| **humidity** | humidity level (%) |
| **wind_speed** | wind speed (km/h) |
| **total_users** | Total number of bike rentals (casual_users + registered_users). This is the target variable |

## Notes:

1. **Evaluation Metrics**: To assess how well your regression model is performing, consider using the following metrics (you will show these metrics in an in-person session).
    - **Mean Squared Error (MSE)**: Measures the average of the squares of errors — the average squared difference between predicted and actual values.
    - **Root Mean Squared Error (RMSE)**: The square root of MSE, providing error in the same unit as the target variable.
    - **R-Squared (R2 Score)**: Indicates the proportion of the variance in the target variable that is predictable from the input features. A higher R2 value means better model performance.
    - **Mean Absolute Percentage Error (MAPE)**: Measures the accuracy as a percentage of the error in predictions.
    - **Mean Absolute Error (MAE)**: Averages the absolute differences between predicted and actual values.
    -
2. **Data Preparation**: In the Kaggle competition, you will need to submit your predictions in a CSV file that includes two columns:
    - `id`: The unique identifier for each record (starting from 1).
    - `label`: The predicted value of `total_users` for each record.

    Make sure your predictions match the required format before submission.

3. **Feature Selection**: Besides correlation analysis, it is highly recommended to use **p-values** for feature selection. This statistical test helps determine which features are significantly related to the target variable. By doing so, you can filter out irrelevant or weakly correlated features that may negatively affect model performance.

4. **Data Augmentation:** Since the dataset contains real-world data, you may also consider using additional data sources to enhance your model. For example, you could retrieve external information about specific days (e.g., holidays, events, or special weather conditions) that might influence bike rental patterns. Although it's not required, leveraging external data could potentially improve prediction accuracy.

Make sure to experiment with different regression algorithms (e.g., Linear Regression, Decision Trees, Random Forest, or Gradient Boosting) and fine-tune the hyperparameters to achieve the best performance on the training data.

To participate in the competition related to this task Click on the competition link: Kaggle Competition.

Good luck with the competition!

# Task 3 : Recommender System

In this task, your goal is to design and evaluate a movie recommendation system that predicts the ratings a user would give to a movie they have not yet rated. You will need to develop a recommendation system using traditional machine learning algorithms (excluding deep learning models) to predict the ratings in the test set.

## Dataset Overview:

- train_data_movie_rate.csv: Contains user ratings for movies with the following columns:
  - user_id: Unique identifier for each user who provided ratings
  - item_id: Unique identifier for each movie that was rated
  - label: The numerical rating given by the user to the movie (rating value)

- train_data_movie_trust.csv: Represents trust relationships between users with the following columns:
  - user_id_trustor: Unique identifier for the user who is expressing trust
  - user_id_trustee: Unique identifier for the user who is being trusted
  - trust_value: A numerical value representing the level of trust
- test_data.csv : This dataset contains pairs of (user_id, item_id) for which your model needs to predict the expected rating that the user would give to the movie.

## Data Exploration and Preprocessing

- Load and explore the movie_rate.txt and movie_trust.txt datasets.
- Check for missing or inconsistent data and handle them if necessary.
- Normalize the ratings if needed (e.g., scaling ratings between 0 and 1).

## Building the Recommender System

- Develop a recommendation system using machine learning algorithms.
- Train the model and evaluate its performance using RMSE, MAE, MSE, R2 as metrics

## How to Submit Your Results to Kaggle

- Your submission should follow the format where you:
  - Read the input file (test_data.csv) containing user_id and item_id pairs
  - Generate predictions for each pair
  - You are required to submit a CSV file with the predicted ratings for each user-item pair in the test set.

- The CSV file should have two columns: `id` (user-item pair) and `label` (predicted rating).

To participate in the competition related to this task Click on the competition link: Kaggle Competition.

Good luck with the competition!

## Evaluation Guidelines:

- This assignment has a total of 100 points: 60 points are allocated to your project submission and its delivery (as reviewed by the TA team), and 40 points are based on your ranking in the contest. (you can earn up to 10 points bonus)
- The contest leaderboard ranking determines how the 40 contest points are awarded. The top group receives 10% of the bonus points, the second group 9%, the third group 8%, and so on, down to the tenth group which receives 1%. Participants ranked beyond the tenth group (i.e., 11th place and onward) will lose 2% of the bonus points for each additional rank. For example, the group in 11th place will receive 38 out of 40 points.
- You will also be required to demonstrate your code live in an in-person session. During this session, your model must produce exactly the same predictions as those submitted to Kaggle.
- If your model's outputs during the live review differ from your Kaggle submission, you will receive a penalty of -100.
- If the uploaded predictions were not directly generated by your model (e.g., if you manually adjusted the results), the penalty will also be -100.
- In regression tasks, Mean Squared Error (MSE) is used for leaderboard ranking.
- In the classification task, Accuracy is used for leaderboard ranking.
- However, in the in-person session, you will be required to report and justify other relevant metrics for each task (e.g., MAE, $R^2$ for regression; Precision, Recall, F1-score for classification). So make sure your code computes and logs these additional evaluation metrics.

## Notes

- Upload your work as a zip file in this format on the website: DS_CA3_[Std number].zip. If the project is done in a group, include all of the group members' student numbers in the name.
- Only one member must upload the work if the project is done in a group.
- We will run your code during the project delivery, so make sure your results are reproducible.
- Only traditional machine learning models are allowed; the use of any neural network models to solve the problem is strictly prohibited.