# Machine Learning – Assignment 1

# Due Date: 11.04.2022

**Data**

The data consists of features of real estate in different areas of Bangalore. It was pre-processed for convenience. The original data can be found here.

Variables:

- **availability**: is the property available immediately (1) or in the near future (0).
- **total_sqft**: the area of the property in square feet (1 foot = 30.54 cm).
- **bedrooms**: the number of bedrooms in the property.
- **bath**: the number of bathrooms in the property.
- **balcony**: the number of balconies in the property.
- **rank**: the ranking of the neighborhood in terms of average price (1 is the highest).
- **area_type**: is the property type a built up area (B) or plot area (P).
- **price in rupees**: the price of the property.

Split:

- **Train**: rows 1-8040.
- **Validation**: rows 8041-10050.
- **Test**: rows 10051-12563.

**Section A (Coding)** *50 pts*

1. Decision Tree: Implement a **Decision Tree** (classifier and regressor) algorithm in Python.
2. AdaBoost: Implement an **AdaBoost** (classifier) algorithm in Python.

**Section B (Implementation)** *30 pts*

1. Classification: Use **both** models from section A and predict the **area type (B, P)**, using all the features in the dataset.
2. Regression: Use the decision tree model from section A and predict the **price** of a property, using all the features in the dataset.

**Section C (Sklearn)** *20 pts*

1. Sklearn Models: Implement the models (including hyperparameter tuning) from section B using built-in function from Sklearn.
2. Comparison: Compare the result of your program and the built-in Sklearn models in terms of metrics and runtime. If there are differences, suggest an explanation.

**Section D (Bonus)** *20 pts*

1. Gradient Boost Regressor:
    - Implement a **Gradient Boost Regressor** algorithm in Python.
    - Run the gradient boost algorithm on the given data to predict the **price** of a property.
    - Compare the algorithm's performance to the built-in Sklearn model and the previous models that you implemented.
2. Classification Metrics:
    - Report the **sensitivity** and **specificity** metrics of section B(1).
    - Is there a significant difference between the scores? Suggest an explanation to why that may be the case.
    - Suggest and apply a method to improve the scores.
3. Performance:
    - Additional bonus points (up to 5) will be given for outperforming other students (in terms of metrics). Make sure to provide an explanation.

**Guidelines**

- Each program should build a model based on the training and validation data.
- Each program should predict the label of the test data, and report the following measures (Sklearn built-in functions allowed):
  - **Accuracy** for classification.
  - **MSE** for regression.
- Impurity measures are **Gini** for classification and **SSR** for regression.
- The implementation should reflect the effect of tuning parameters.
- **Do not use** the Sklearn library or any other explicit machine learning libraries unless clearly stated otherwise.
- Try to minimize the usage of loops, lists and other inefficient programing habits. Numpy library has a lot of useful built-in function. In this case, Google is your best friend.

**Submission**

- The assignment should be submitted in pairs (only one submission).
- You are required to submit two files including all the sections. One in **.ipynb** format and one in **.html**. Both files should also include the program's outputs.
- The files' names should be of the form: **ML_HW1_#ID1_#ID2**.
- Assignments submitted late will receive a penalty of **3 points** for each day, up to one week. Later submissions will not be accepted.