Group B - No-Tool Experiment Tasks

You will perform the tasks with the python file (notool-b.py) which contains the solution to a regression problem, predicting housing prices in Boston. This has been done using SK-Learn. You will perform the following instructions and tasks where you will alter certain data such as hyperparameter and dataset features, and run the regression using two different supervised learning algorithms: Linear_regression and Random Forest Regressor algorithms.

1) You are required to track the following assets/data using the way you see best:

- The datasets and features used for each run
- Relevant parameters and hyper-parameters are used in each run.
- The model generated in each run
- The evaluation metrics obtained from generated model

Experimenting with algorithm 1 (Linear Regression)

Carry out several experiment runs using different sets of dataset features and learning parameters as described below. Execute the python script to train and evaluate the model.

1. **Run 1**: Use the default dataset features and parameter values and execute the python scripts:

   $ python notool-b.py

2. **Run 2**: Change the normalize parameter to False and execute the script again:

   $ python notool-b.py

3. **Run 3**: In the Boston dataset we have a total of 13 features::

```
CRIM: Per capita crime rate by town
ZN: Proportion of residential land zoned for lots over 25,000 sq. ft
INDUS: Proportion of non-retail business acres per town
CHAS: Charles River dummy variable (= 1 if tract bounds river; 0
otherwise)
NOX: Nitric oxide concentration (parts per 10 million)
RM: Average number of rooms per dwelling
AGE: Proportion of owner-occupied units built prior to 1940
DIS: Weighted distances to five Boston employment centers
RAD: Index of accessibility to radial highways
TAX: Full-value property tax rate per $10,000
PTRATIO: Pupil-teacher ratio by town
B: 1000(Bk − 0.63)², where Bk is the proportion of [people of African
American descent] by town
LSTAT: Percentage of lower status of the population
MEDV: Median value of owner-occupied homes in $1000s
```

Now we will train the model with 3 features:'PTRATIO', 'TAX', 'RM'. Uncomment line 24.

Run the file again: *$ python notool-b.py*

4. **Run 4**: We have used 80% (0.20) of the dataset as a training set. Change the test_size to 70% (0.30) That's in split_param, for a change.

```
# Splitting the data and setting test_size
split_param = {'test_size': 0.20, 'random_state': 28750}

X_train, X_test, y_train, y_test = train_test_split(data, y, **split_param)
```

Run the script again: *$ python notool-b.py*

5. **Run 5:** We will now use other features to train the model. Change the 'PTRATIO' to 'CRIM'. Also, set the "normalize" parameter to True

Run the file again: *$ python notool-b.py*

6. **Run 6:** Use all features from the dataset to train the model. Do this by commenting out line 24.

Run the file again: *$ python notool-b.py*

7. **Run 7:** Add '**B**' to the features in line 24. And uncomment line 24 again.

Run the file again: *$ python notool-b.py*

Experimenting with algorithm 2 (RFR)

We have carried out a series of runs using the Linear Regression algorithm, and now we will use the RFR algorithm. This will give us different results compared to the previous algorithm.

1. **Run 8:** Comment out the Linear regression model with (#), and uncomment the RandomForestRegressor, as shown below: (Lines 38-39)

```
model = RandomForestRegressor(**parameters)
#model = LinearRegression(normalize=True)
```

   Run the file again: *$ python notool-b.py*

2. **Run 9**: The parameter we are using for the RFR is in line 32, uncomment it.

```
# The parameters for Random Forest Regressor
parameters_rfr = {'n_estimators': 50, 'max_depth': 5, 'min_samples_split': 6, 'ccp_alpha'=0.1}
```

   Run the script again: *$ python notool-b.py*

3. **Run 10**: To avoid overfitting we will change the ccp_alpha. Change the ccp_alpha variable to 0.01

   Run the script again: *$ python notool-b.py*

4. **Run 11**: Now we will add more features to train the model. Add 'LSTAT' in feature selection. This is how it should look:

```
# Feature Selection
df = df[['CRIM', 'TAX', 'RM', 'AGE', 'B', 'LSTAT']]
```

   Run the script again: *$ python notool-b.py*

5. **Run 12**: Let's try to improve the model. The default value of max_features is None. Add the following to the parameter dictionary in line 32:

   'max_features': 'log2'

   Run the file again: *$ python notool-b.py*

6. **Run 13**: Change the test_size back to 0.20.

   Run the file again: *$ python notool-b.py*

*Now return to the experiment's questionnaire