Computer Science Department
CS660 – Mathematical Foundations of Analytics (CRN# 71425)
Fall 2024

Project #3 / Due 09-Dec-2024

Let's review how the **CART** (**C**lassification **A**nd **R**egression **T**ree) algorithms work in Python's scikit-learn module.
Namely, we should study the following **Decision Tree** libraries:

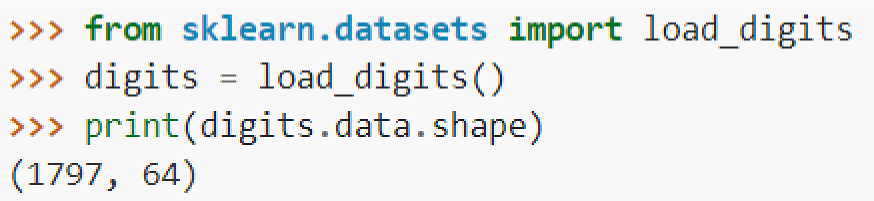
1_ DecisionTree**Classifier**
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

2_ DecisionTree**Regressor**
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

3_ RandomForest**Classifier**
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

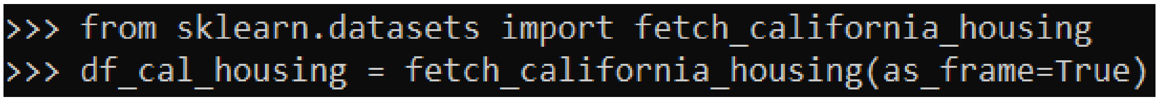
4_ RandomForest**Regressor**
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

You should perform both Classification and Regression modeling on the following datasets:
A_ For **classification** use scikit-learn's hand-written digits (each data point is an 8X8 image of a single digit)
https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html#sklearn.datasets.load_digits



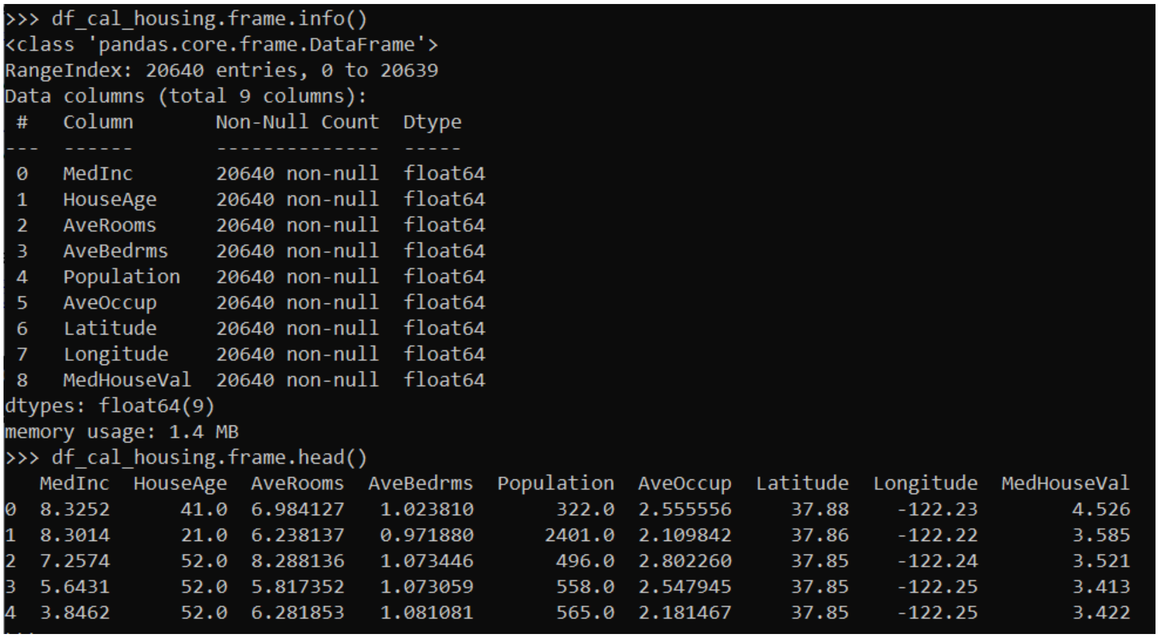
```
>>> from sklearn.datasets import load_digits
>>> digits = load_digits()
>>> print(digits.data.shape)
(1797, 64)
```

B_ For **regression** use scikit-learn's California-housing dataset
https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html#sklearn.datasets.fetch_california_housing



```
>>> from sklearn.datasets import fetch_california_housing
>>> df_cal_housing = fetch_california_housing(as_frame=True)
```

The target feature is 'MedHouseVal'. This is the value of a house; you need to predict it.



```
>>> df_cal_housing.frame.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   MedInc       20640 non-null  float64
 1   HouseAge     20640 non-null  float64
 2   AveRooms     20640 non-null  float64
 3   AveBedrms    20640 non-null  float64
 4   Population   20640 non-null  float64
 5   AveOccup     20640 non-null  float64
 6   Latitude     20640 non-null  float64
 7   Longitude    20640 non-null  float64
 8   MedHouseVal  20640 non-null  float64
dtypes: float64(9)
memory usage: 1.4 MB
>>> df_cal_housing.frame.head()
   MedInc  HouseAge  AveRooms  AveBedrms  Population  AveOccup  Latitude  Longitude  MedHouseVal
0  8.3252      41.0  6.984127   1.023810       322.0  2.555556     37.88    -122.23        4.526
1  8.3014      21.0  6.238137   0.971880      2401.0  2.109842     37.86    -122.22        3.585
2  7.2574      52.0  8.288136   1.073446       496.0  2.802260     37.85    -122.24        3.521
3  5.6431      52.0  5.817352   1.073059       558.0  2.547945     37.85    -122.25        3.413
4  3.8462      52.0  6.281853   1.081081       565.0  2.181467     37.85    -122.25        3.422
```

Write **Python/R** scripts (within a Notebook) in order to complete the following tasks:

1_ Perform Exploratory Data Analysis (EDA) on both datasets.

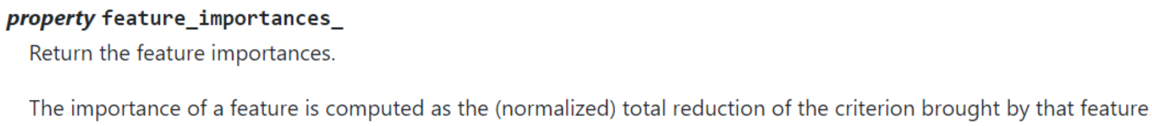
2_ Perform classification task on the digit's dataset, utilizing both DTClassifier and RandomForest with their default parameters. Which one performs better? Print out their respective Confusion Matrices.

3_ Perform same task as #1, but now tune the classifiers. Which one performs better? Print out their respective Confusion Matrices.

4_ Perform regression analysis task on the California-housing dataset, utilizing both DTRegressor and RandomRegressor with their default parameters. Which one performs better? Print out their respective Mean-Squared Error (MSE) and Coefficient of Determination (R-squared, R^2)

5_ Perform same task as #3, but now tune the regressors. Which one performs better? Print out their respective Mean-Squared Error (MSE) and Coefficient of Determination (R-squared, R^2)

6_ Print out, for each of the tasks above, the feature importance (aka Gini importance) for each of the features in the dataset.



property `feature_importances_`

Return the feature importances.

The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature